

**Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství**

**Platforma pro měření neelektrických veličin prostřednictvím
Raspberry Pi a IQRF – laboratorní model**

**Platform for Measuring of Non-electrical Quantities through the
Raspberry Pi and IQRF – Laboratory Model**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání bakalářské práce

Student: **Daniel Kadura**
Studijní program: **B2649 Elektrotechnika**
Studijní obor: **2612R041 Řídicí a informační systémy**
Téma: **Platforma pro měření neelektrických veličin prostřednictvím Raspberry Pi a IQRF - laboratorní model**
Platform for Measuring of Non-electrical Quantities through the Raspberry Pi and IQRF - Laboratory Model
Jazyk vypracování: **čeština**

Zásady pro vypracování:

1. Seznámení se s IQRF technologií a DPA protokolem.
2. Seznámení se s realizovanými bezdrátovými měřicími moduly.
3. Implementace IQRF SDK pro komunikaci mezi Raspberry Pi a IQRF.
4. Návrh algoritmu pro vyčítání dat z bezdrátových snímačů a jejich obsluhy.
5. Vytvoření archívy a vizualizace měřených dat.
6. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] MICRORISC. *IQRF Quick Start Guide For IQRF OS v3.07D and higher*. 2017. [online] Dostupné z: <http://www.iqrf.org>.
- [2] MICRORISC. *IQRF OS Operating System Version 4.00D for (DC)TR-7xD - User's Guide*. 2017. [online] Dostupné z: <http://www.iqrf.org>.
- [3] MICRORISC. *IQRF DPA Framework Technical Guide v3.00*. 2017. [online]. Dostupné z: <http://www.iqrf.org>.
- [4] Firemní dokumentace MICRORISC. *HW Profile 0x12 for cbe.vsb.cz*. 2015.
- [5] PIEŠ, Martin et al. *Dokumentace k modifikovaným verzím bezdrátových snímačů*. Dílčí zpráva projektu MPO č. FR-TI4/327, 2016.


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Martin Pieš, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2019




doc. Ing. Jiří Koziolek, Ph.D.
vedoucí katedry


prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 29. dubna 2019

A handwritten signature in blue ink, appearing to read 'Michal', is written over a horizontal dotted line.

Podpis studenta

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce Ing. Martinu Piešovi, Ph.D za odbornou pomoc, konzultaci a cenné rady při vytváření této bakalářské práce.

Dále bych chtěl poděkovat mému nejlepšímu kamarádovi Filipu Gawlasovi za to, že mi byl během studia oporou.

Abstrakt

Tato bakalářská práce se zabývá návrhem laboratorního modelu pro bezdrátové měření neelektrických veličin, konkrétně vzdálenosti, jejíž měření je realizováno pomocí ultrazvukového senzoru SFR08, teplota a relativní vlhkost je měřena senzorem SHT21 a intenzita osvětlení je měřena fotorezistorem.

Bezdrátový přenos dat zajišťuje IQRF modul řady TR-72D společnosti Microrisc. Další zpracování naměřených dat je zajištěno pomocí Raspberry PI a jejich následná vizualizace v programu Node-RED.

V teoretické části práce jsou rozebrány detailně všechny výše vyjmenované veličiny společně se senzory pro jejich měření včetně samotného IQRF modulu, který byl použit.

Klíčová slova

IQRF, teplota, relativní vlhkost, vzdálenost, intenzita osvětlení, senzor, měření

Abstract

This bachelor thesis deals with the design of a laboratory model for wireless measurement of non-electrical quantities, namely the distance measured using the ultrasonic sensor SFR08, the temperature and relative humidity are measured by the SHT21 sensor and the intensity of the light is measured by the photoresist.

Wireless data transmission is provided by the Microrisc TR-72D IQRF module. Next processing of measured data is via Raspberry PI and their subsequent visualization in the Node-RED.

In the theoretical part of the thesis are analyzed in detail all the above mentioned quantities together with sensors for their measurement including the IQRF module itself, which was used.

Key words

IQRF, temperature, relative humidity, distance, intensity of the light, sensor, measurement

Obsah

Seznam použitých symbolů a zkratk	8
Seznam obrázků:	11
1 Úvod.....	13
2 Měření veličin	14
2.1 Teplota.....	14
2.1.1 Kontaktní měření teploty.....	14
2.1.2 Bezkontaktní měření teploty	15
2.2 Poloha.....	16
2.2.1 Snímače polohy	16
2.3 Intenzita osvětlení	19
2.3.1 Měření intenzity osvětlení	20
3 IQRF.....	24
3.1 Úvod do IQRF	24
3.2 Technické parametry IQRF	24
3.3 Transceiver TR-72D.....	24
3.4 IQMESH	26
3.5 Aplikační přístupy	27
3.5.1 Bez programování (DPA).....	27
3.5.2 S programováním v jazyce C	27
4 Návrh a realizace modulů.....	28
4.1 Návrh rozhraní mezi senzory a DK-EVAL-04A.....	28
4.1.1 Modul SFR08-EVAL	28
4.1.2 Modul SHT21-EVAL.....	30
4.1.3 Modul PHOTORES-EVAL.....	31
5 Vytvoření IQRF sítě.....	32
6 Vytvoření Custom DPA handleru	34
6.1 Custom dpa handler pro SFR08	34
6.2 Custom dpa handler pro SHT21	35
6.3 Custom dpa handler pro Fotorezistor	37
6.4 DPA Autoexec	38
7 IQRF Gateway	39
8 Node-RED.....	43
8.1 Poslání Requestu	43
8.2 Přijmutí response.....	44
8.3 Vizualizace.....	45

8.4	Ukládání dat	47
8.5	Ověření připojení.....	48
8.6	Napětí nodů	49
8.7	Teplota CPU Raspberry PI.....	50
9	Závěr	51
	Zdroje	53
	Seznam příloh:	55

Seznam použitých symbolů a zkratek

Zkratka	Význam
A/D	Převod analogového signálu na digitální
CdS	Sulfid kademnatý
CdSe	Selenid kademnatý
CdTe	Telurid kademnatý
CPU	Centrální procesorová jednotka
ČSN	Česká technická norma
DPA	Přímý přístup k perifériím
E	Značka intenzity osvětlení
EEPROM	Elektricky mazatelná paměť, pro zápis i čtení
HTTP	Internetový protokol
HWPID	Identifikátor hardwarového profilu
I	Značka elektrického proudu
I2C	Inter-Integrated circuit, počítačová sériová sběrnice
I/O	Vstup, výstup
IoT	Internet věcí
IQMESH	MESH síť
IQRF	Technologie pro bezdrátový přenos dat
K	Kelvin, základní jednotka teploty

LED	Light-Emitting Diode, dioda emitující světlo
lm	Základní jednotka intenzity osvětlení, lumen
LP	Low-power, režim nízké spotřeby
lx	Základní jednotka světelného toku, lux
m	Metr, základní jednotka vzdálenosti
m²	Základní jednotka plochy, metr čtvereční
MHz	Megahertz, milion hertzů
MQTT	Komunikační protokol
mW	Miliwatt, tisícina wattu
NADR	Číselné označení modulu v síti
OS	operační systém
PCMD	Příkaz pro danou periférii
PDATA	Datová zpráva pro daný modul
PHP	Skriptovací programovací jazyk
PN	Přechod polovodiče typu P (pozitivní) a polovodiče typu N (Negativní)
PNUM	Číselné označení periferie daného modulu
RF	Radio frequency, Rádiový signál
RX	Příjem
s	Vzdálenost
S	Značka plochy

SPI	Serial Peripheral Interface, sériové periferní rozhraní
STD	Standartní režim
t	Čas
T₁	Teplota v čase 1
T₂	Teplota v čase 2
TR-52D	Označení modulu, starší typ
TR-72D	Označení modulu, novější typ
TX	Vysílání
U	Značka elektrického napětí
USB	Universal Serial Bus, univerzální sériová sběrnice
v	Rychlost
V	Základní jednotka napětí, volt
XLP	Extrem Low-power, režim extrémní nízké spotřeby
μA	Mikroampér, miliontina ampéru
Φ	Značka světelného toku
Ω	Ohm, základní jednotka odporu
°C	Stupeň Celsia, základní jednotka Celsiovy stupnice
°F	Stupeň Fahrenheita, základní jednotka Fahrenheitovy stupnice

Seznam obrázků:

<i>Obr. 1 - Závislost tělesa a vlnové délky jim vyzařovaného [4]</i>	15
<i>Obr. 2 - Konstrukce bolometru [3]</i>	15
<i>Obr. 3 - Graf intenzity vyslané a odražené vlny [11]</i>	18
<i>Obr. 4 - Blokové schéma snímače</i>	18
<i>Obr. 5 - Snímač SFR08 s I2C sběrnici [13]</i>	19
<i>Obr. 6 - Odrazový vzor SFR08 [13]</i>	19
<i>Obr. 7 - Průřez hradlového fotočlánku [14]</i>	20
<i>Obr. 8 - Schéma hradlového fotočlánku [14]</i>	20
<i>Obr. 9 - Závislost velikosti proudu na světelném toku [14]</i>	21
<i>Obr. 10 - Příklad luxmetru EXTECH LT300 [14]</i>	21
<i>Obr. 11 - Schématická značka fotorezistoru [15]</i>	22
<i>Obr. 12 - Závislost odporu na osvětlení (CdS) [15]</i>	22
<i>Obr. 13 - Voltampérová charakteristika fotorezistoru [16]</i>	22
<i>Obr. 14 - Ukázka fotorezistoru [17]</i>	23
<i>Obr. 15 - Transciever TD-72 s blokovým schématem [19]</i>	25
<i>Obr. 16 - Zjednodušené vnitřní zapojení TR-72D [19]</i>	26
<i>Obr. 17 - Demonstrace IQMESH sítě</i>	26
<i>Obr. 18 - Blokové schéma modelu bez programování a s programováním [19]</i>	27
<i>Obr. 19 - Schéma zapojení SFR08-EVAL</i>	28
<i>Obr. 20 - Ukázka modulu SFR08-EVAL</i>	29
<i>Obr. 21 - Schéma zapojení SHT21-EVAL</i>	30
<i>Obr. 22 - Ukázka modulu SHT21-EVAL</i>	30
<i>Obr. 23 - Schéma zapojení PHOTORES-EVAL</i>	31
<i>Obr. 24 - Ukázka modulu PHOTORES-EVAL</i>	31
<i>Obr. 25 - Vytvořená IQRF MESH síť</i>	32
<i>Obr. 26 - Ukázka vyčtení dat ze senzoru SFR08</i>	34
<i>Obr. 27 - Princip vyčtení dat ze senzoru SHT21</i>	36
<i>Obr. 28 - Princip fungování programu pro vyčtení hodnoty z fotorezistoru</i>	37
<i>Obr. 29 - Paket pro nastavení DPA hodnoty koordinátora sítě</i>	38
<i>Obr. 30 - Ukázka KON-RASP-01</i>	39
<i>Obr. 31 - Struktura návrhu</i>	39
<i>Obr. 32 - Ukázka Webapp včetně konfiguračních souborů</i>	40
<i>Obr. 33 - Zobrazení připojeného koordinátora sítě</i>	41
<i>Obr. 34 - Blokové schéma komunikace</i>	42
<i>Obr. 35 - Řetězec pro posílání requestu v programu Node-RED</i>	43
<i>Obr. 36 - Řetězec pro přijetí odpovědi v programu Node-RED</i>	44
<i>Obr. 37 - Řetězec pro vizualizaci dat v programu Node-RED</i>	45
<i>Obr. 38 - Celková vizualizace měřených dat v programu Node-RED</i>	47
<i>Obr. 39 - Ukázka návrhu tabulky v phpMyadmin</i>	48
<i>Obr. 40 - Řetězec pro ukládání dat do databáze v programu Node-RED</i>	48
<i>Obr. 41 - Použití STATUS1 v programu Node-RED</i>	49
<i>Obr. 42 - Použití STATUS2 v programu Node-RED</i>	49
<i>Obr. 43 - Použití STATUS3 v programu Node-RED</i>	49

<i>Obr. 44 - Ukázka řetězce pro vyčtení stavu baterie nodů</i>	<i>50</i>
<i>Obr. 45 - Vyčtení teploty CPU Raspberry PI v programu Node-RED.....</i>	<i>50</i>
<i>Obr. 46 - Vytvořený panel.....</i>	<i>51</i>
<i>Obr. 47 - Přihlášení a zvolení databáze v aplikaci phpMyAdmin.....</i>	<i>1</i>

1 Úvod

V dnešní době je dálkové měření neelektrických veličin velmi rozšířené a existuje proto nespočet aplikací a zařízení, pomocí kterých ho můžeme realizovat. Tato metoda je rozšířená zejména v průmyslu, kde je potřeba měřit nepřeborné množství elektrických, a právě i neelektrických veličin, které značně převládají, a hlavně je možno mít z celého komplexu data o měřených veličinách na jednom místě s možností jejich dalšího zpracování. Měření neelektrických veličin probíhá jejich převodem na elektrický signál, který je dále zpracováván pro potřeby jednotlivého uživatele.

Jednou z metod pro vzdálené měření neelektrických veličin je pomocí IQRF modulu, ke kterému je vhodně připojený příslušný senzor pro měřenou veličinu, jakou potřebujeme v daný okamžik měřit. Tato měření jsou vhodná zejména proto, že naměřená data mohou být dále zpracována pro potřeby uživatele. Pro tuto bakalářskou práci byly zvoleny jako neelektrické veličiny vzdálenost, teplota a intenzita osvětlení. Tyto veličiny byly pro tuto práci stanoveny na základě toho, že se jedná v praxi o nejrozšířenější měřené neelektrické veličiny vůbec.

Měření teploty je možno provádět dvěma způsoby, a to je kontaktně, kdy senzor přichází do přímého styku s měřeným objektem nebo bezkontaktně neboli pyrometricky, kdy snímač dokáže změřit teplotu tělesa na základě toho, jak velké elektromagnetické záření těleso vyzařuje.

Měření polohy bylo realizováno pomocí ultrazvukového senzoru, který patří mezi nejrozšířenější senzory v tomto odvětví.

A na závěr měření intenzity osvětlení, kde je hodnota světelné intenzity převedena pomocí fotorezistoru na hodnotu napětí, s kterou je možno dále pracovat.

2 Měření veličin

2.1 Teplota

Teplota je stavová veličina, patřící mezi základní veličiny SI soustavy, která popisuje stav hmoty, jejíž základní jednotka je Kelvin (K). Mezi vedlejší stupnice dále patří Celsiova stupnice, která má základní jednotku stupeň Celsia ($^{\circ}\text{C}$) a Fahrenheitova stupnice se základní jednotkou stupeň Fahrenheita ($^{\circ}\text{F}$). V technice je teplota důležitá zejména pro definování pracovních podmínek jednotlivých zařízení či k měření teplot v průmyslu pro vybavení jako jsou vysokoteplotní pece a podobně. [1]

2.1.1 Kontaktní měření teploty

Kontaktní měření teploty se používá v místech, kde je možnost dostat teplotní senzor do přímé blízkosti měřeného objektu. Základem je, aby teplo bylo co nejlépe sdíleno mezi senzorem a měřeným povrchem. Senzor by měl mít co nejmenší hmotnost a pokud možno co největší dotykovou plochu. Dále by měl senzor mít stejné emisní vlastnosti jako povrch měřeného tělesa. [2]

Odporové senzory teploty

Měření teploty pomocí odporových senzorů je založeno na teplotní závislosti odporu daného materiálu, jehož volbou určujeme přesnost, teplotní rozsah, popřípadě samotnou konstrukci senzoru. Důraz je kladen na to, aby teplotní součinitel odporu daného materiálu byl co největší, stálý, teplotní závislost odporu co nejlineárnější a jeho rezistivita co největší. Proud, jenž protéká senzorem, by měl být co nejmenší, aby jeho průchod způsoboval co nejmenší možné oteplení a nezkresloval tak naměřené hodnoty. Podle materiálu můžeme rozdělit odporové senzory teploty na kovové a polovodičové, které v této době patří mezi nejvíce propracované a mají čím dál větší význam. Můžeme je dále rozdělit na polykrystalické a monokrystalické. [2]

Termoelektrické senzory

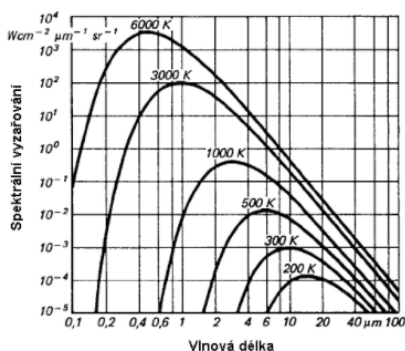
Jsou to senzory, využívající Seebeckova jevu. To znamená, že pokud jsou spojeny dva vodiče z různých kovů do uzavřeného obvodu a mají-li spoje různou teplotu T_1 a T_2 , protéká obvodem elektrický proud. Pokud obvod rozpojíme, na svorkách naměříme elektromotorické napětí. Jsou vhodné zejména pro kontaktní měření vyšších teplot a pro měření rychlých změn teplot, jelikož mají malou teplotní setrvačnost. [3]

Bimetalové teploměry

K měření teploty je využit bimetalový pásek, který se skládá ze dvou kovů, kde každý z nich má jiný činitel tepelné roztažnosti. Při změně teploty se pásek deformuje a na základě deformace jsme schopni určit teplotu.

2.1.2 Bezkontaktní měření teploty

Bezkontaktní měření teploty neboli pyrometrie, je způsob, jak měřit teplotu na hůře dostupných místech, u pohyblivých objektů nebo u předmětů, kde nesmí být měřením poškozena či nějak jinak dotčena. Využívá se principu, kdy zahřátý objekt vyzařuje do okolí infračervené záření. Můžeme takto měřit pouze teplotu povrchu měřeného objektu, a to pouze ve směru, kterým je senzor namířen. Jedná se velmi rychlé měření, které se pohybuje v řádu milisekund s velkým rozsahem teplot, které mohou dosáhnout teploty až 3000 °C. Nevýhodou je vnikající chyba, způsobena špatným nastavením emisivity měřeného objektu a prostupností prostředí mezi senzorem a měřeným objektem. [4]



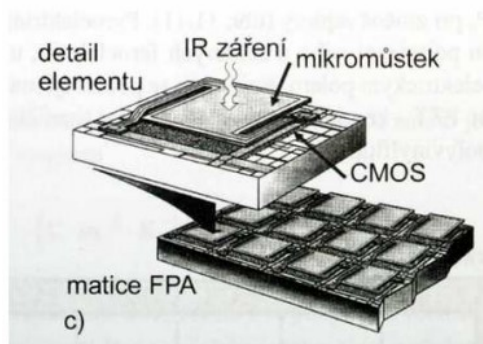
Obr. 1 - Závislost tělesa a vlnové délky jim vyzařovaného [4]

Infračervené termočlánky

Snímací část senzoru je načerněna, čímž dochází k absorpci infračerveného záření, které způsobuje jeho zahřívání. Teplota zahřátí snímací části senzoru se poté snímá termočlánky.

Bolometry

Princip je stejný jako u Infračervených termočlánků, jen při zahřívání snímací části senzoru se využívá termokonduktivní jev, to znamená, že změnou teploty na povrchu senzoru se převede na změnu vodivosti. [4]



Obr. 2 - Konstrukce bolometru [3]

Pyrometrické senzory:

Jedná se o senzory pracující na pyroelektrickém jevu, to znamená, že pokud je těleso vystaveno vlivu změny teploty, jeho povrch se nabíjí spontánní polarizací. Pokud poté na těleso dopadne infračervené záření, tak v tom místě vznikne elektrický náboj, jehož velikost je možno dále změřit. Tento jev mají pyroelektrika s permanentní polarizací a společně s některými feroelektriky, u kterých je orientace vytvořena silným elektrickým polem. [4]

Kvantové senzory infračerveného záření:

Senzory, jež využívají fyzikálních jevů, které vznikají při přímém vzájemném působení dopadajících fotonů se senzorem. Senzor využívá polovodičových fotodiod a fotovodivostních detektorů, které jsou nastaveny pro infračervenou oblast a zároveň nastaveny nejčastěji do napětíového režimu. Senzor je třeba chladit z důvodu vysokých nároků na odstup signál – šum. [4] [5]

2.2 Poloha

Jedná se o označení umístění nebo orientaci námi snímaného tělesa vzhledem k okolnímu prostředí. Nejčastěji se udává v metrech (m). Mezi senzory polohy se řadí rovněž snímače přítomnosti, které mají své uplatnění zejména v průmyslu.

2.2.1 Snímače polohy

O odporové

O odporové snímače polohy dělíme na dva druhy, a to na spojitě a nespojitě. Spojitě snímače jsou takové, kde elektrické napětí nebo proud jsou přímo úměrné měřené poloze. Pro realizaci se používají potenciometry nebo trimry.

Nespojitě snímače polohy fungují na principu přítomnosti, tedy jestli je věc přítomna nebo ne, výstupem takového snímače je tedy 0 nebo 1. Zde se používají spínače rtuťové, magnetické a mechanické. [6]

Kapacitní

Základní princip kapacitního snímače spočívá ve vyhodnocování polohy, jež je spojeno se změnou kapacity v okruhu snímače nebo změny kapacity kondenzátoru použitého ve snímači. Kapacita je závislá na změně tloušťky, permitivity a plochy dielektrika. Výhodou těchto čidel je, že dokáží reagovat téměř na jakýkoliv materiál. [7]

Indukční

Poloha je vyhodnocena na základě změny vlastní indukčnosti cívky, popřípadě vzájemné indukčnosti. Výhodou těchto snímačů je možnost použití i ve velmi nepříznivých podmínkách, či v prostorách, kde je riziko nebezpečí výbuchu. Jejich nevýhodou je, že dokáží snímat pouze elektricky vodivé předměty. [8]

Optické

Princip snímače spočívá ve vyhodnocení přítomnosti světelného paprsku mezi vysílačem a přijímačem a polohou snímaného předmětu. Poté se vykoná převod na elektrickou veličinu, kterou je možné dále zpracovat. Optické snímače dále dělíme na absolutní, kdy je poloha měřena vzhledem k referenčnímu bodu, inkrementální, kdy je poloha měřena vůči minulému bodu a limitní, kdy je polohu nutné vyhodnotit dvouhodnotově. Výhody optického snímání vzdálenosti je, že díky rychlosti světla a velmi rychlé odezvě snímače je možné určit polohu měřeného objektu téměř v kterémkoliv okamžiku. Nevýhodou je vyšší cena jak snímače samotného, tak jeho příslušenství. [9] [10]

Ultrazvukové snímače polohy

Ultrazvukový snímač pracuje na principu měření doby odrazu ultrazvukových pulsů od měřeného objektu. Poté z naměřené doby a známé rychlosti zvuku, kterou je nutné pro dané podmínky definovat lze určit vzdálenost pomocí: [11]

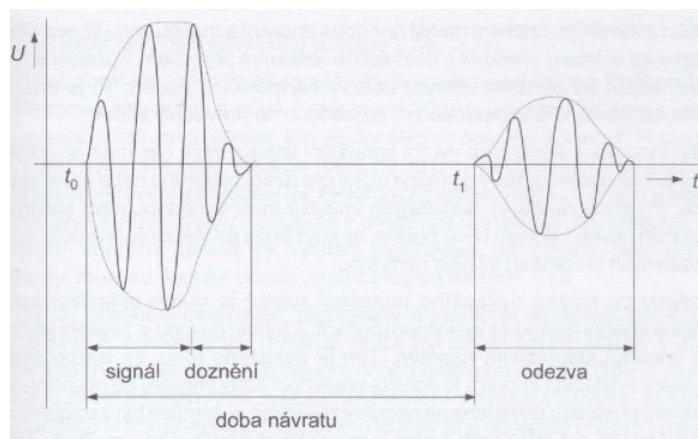
$$s = v \cdot t$$

- v – rychlost zvuku (ms^{-1})
- t – doba odrazu pulsů od předmětu podělená 2 (s)
- s – výsledná vzdálenost (m)

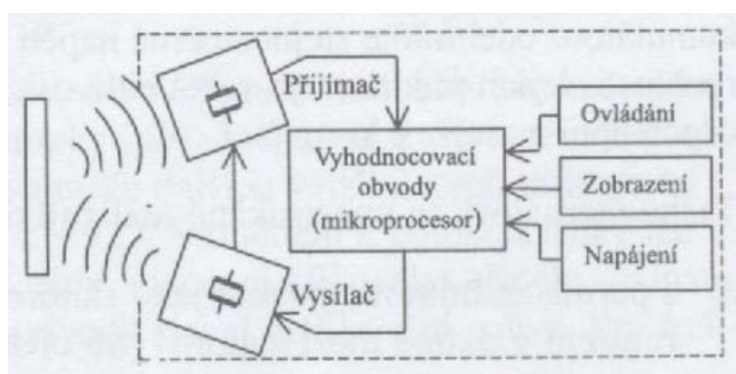
Výpočet rychlosti zvuku, za předpokladu, že měřený objekt je ve vzduchovém prostředí:

$$v = (v_0 + 0,61T)$$

- v – rychlost zvuku (ms^{-1})
- T – teplota prostředí v $^{\circ}\text{C}$
- v_0 – rychlost zvuku při teplotě 0°C ($331,6 \text{ ms}^{-1}$)



Obr. 3 - Graf intenzity vyslané a odražené vlny [11]

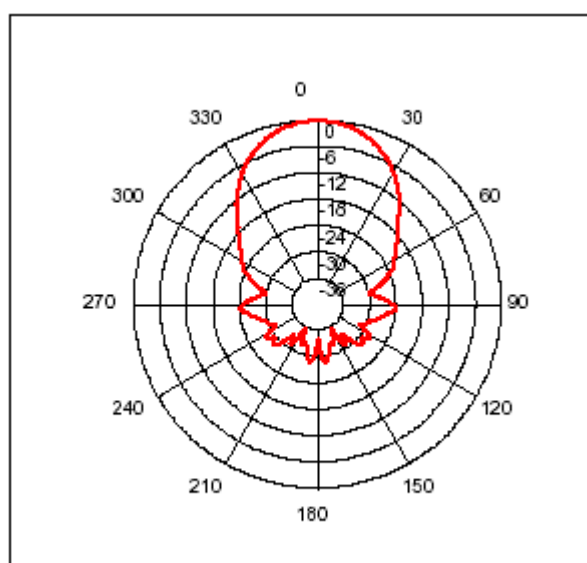


Obr. 4 - Blokové schéma snímače

Je několik typů ultrazvukových snímačů. Jednohlavé systémy, kde je vysílač tvořen přímou kombinací s přijímačem. Dále dvouhlavé (viz obr. 5), kde přijímač je oddělený od vysílače. Mezi jejich výhodu patří možnost měřit jakékoliv předměty, co se do barvy a složení týče. Nevýhodou je, že přesnost měření je ovlivněna vlhkostí a teplotou prostředí. [12]



Obr. 5 - Snímač SFR08 s I2C sběrnici [13]



Obr. 6 - Odrazový vzor SFR08 [13]

2.3 Intenzita osvětlení

Jedná se o fotometrickou veličinu, definovanou jako světelný tok, který dopadá na jednotku plochy. To znamená, že je podílem světelného toku (v lumenech) a plochy (v metrech čtverečních). Intenzitu osvětlení značíme E a jednotkou je lux (lx)

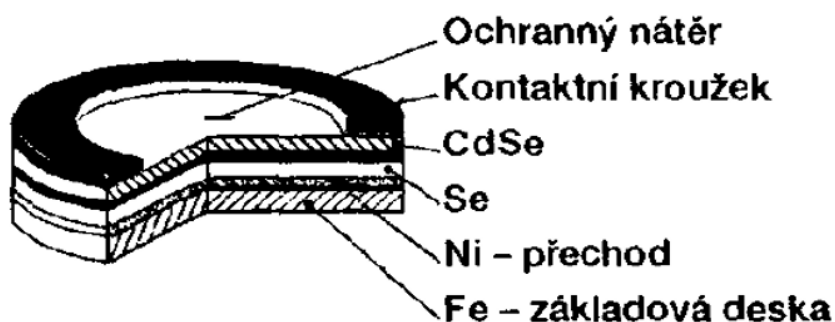
$$E = \frac{d\Phi}{dS}$$

- Φ – Světelný tok (lm)
- S – plocha (m²)
- E – intenzita osvětlení (lx)

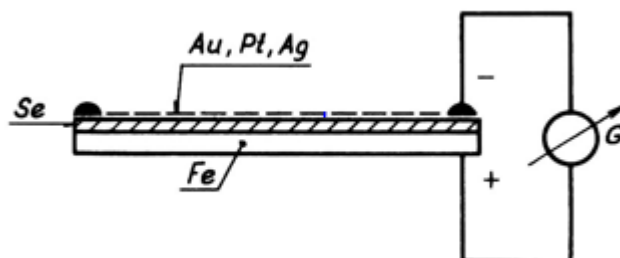
2.3.1 Měření intenzity osvětlení

Luxmetry

Na základní desce, která je vytvořena nejčastěji ze železa (hliníku) je nanesená polovodičová vrstva tvořena křemíkem (dříve selenem). Na polovodičové vrstvě je průsvitná vodivá vrstva tvořena platinou, stříbrem, popřípadě zlatem. Po celém obvodu je sběrný kroužek. Princip luxmetru spočívá v tom, že při ozáření vznikne mezi polovodičovou a vodivou vrstvou rozdíl potenciálů, kdy proud začne procházet sběrným kroužkem, který je dále měřen pomocí galvanoměru.

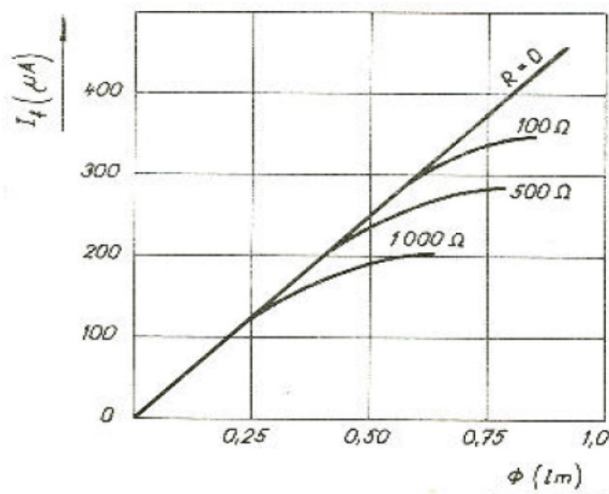


Obr. 7 - Průřez hradlového fotočlánku [14]



Obr. 8 - Schéma hradlového fotočlánku [14]

Pro přesné měření je nutno splnit podmínku, kdy křivka spektrální citlivosti přístroje musí být stejná jako u lidského oka. Pokud tomu tak není, je možné měřit pouze světelný zdroj, na který je luxmetr cejchován, jinak vzniká velká chyba měření.



Obr. 9 - Závislost velikosti proudu na světelném toku [14]

Dále se rozlišuje několik typů přesností, které korespondují s českou normou ČSN 36 0011-1:

- 1) L – laboratorní měření (celková chyba je $\pm 3 \%$)
- 2) A – přesné (celková chyba je $\pm 5 \%$)
- 3) B – provozní (celková chyba je $\pm 10 \%$)
- 4) C – orientační (celková chyba je $\pm 20 \%$)

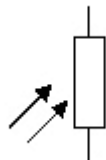


Obr. 10 - Příklad luxmetru EXTECH LT300 [14]

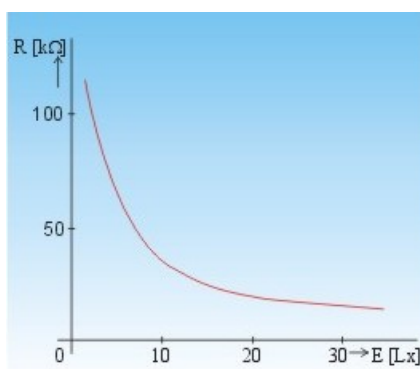
Fotorezistory

Fotorezistor (fotoodpor, viz. obr. 14) je polovodičový jednobran, jenž nemá PN přechod. Je zde využit vnitřní fotoelektrický jev, kdy je odpor součástky závislý na intenzitě osvětlení. Jsou konstruovány tak, aby jejich poměr odporu za tmy byl daleko větší než poměr odporu při osvětlení. Za

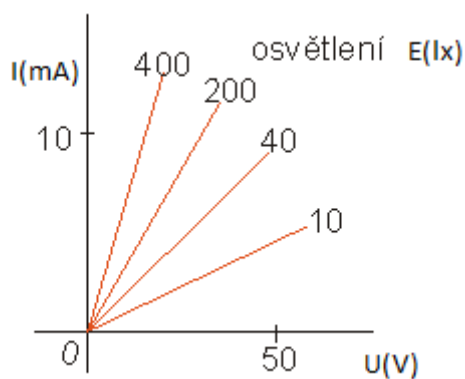
tmy dosahuje odpor součástky hodnoty v intervalu od 10^6 až $10^9 \Omega$. Jejich výroba probíhá tak, že se napaří vrstva vhodného polovodičového materiálu (CdS, CdSe nebo CdTe pro infračervené záření) na podložku z keramiky. Polovodičová vrstva má meandrovitý tvar, aby se dosáhlo zvětšení odporu součástky. Pokud dojde k osvětlení citlivé vrstvy, odpor fotorezistoru se zmenší. [15]



Obr. 11 - Schématická značka fotorezistoru [15]



Obr. 12 - Závislost odporu na osvětlení (CdS) [15]



Obr. 13 - Voltampérová charakteristika fotorezistoru [16]

Odezva fotorezistoru na změnu osvětlení je poměrně pomalá. Odezva se mění s intenzitou osvětlení, která působí na fotorezistor. Zrychluje se s větší velikostí intenzity osvětlení současně s většími vlnovými délkami. Rychlost odezvy také závisí na materiálu, z kterého je fotorezistor vyroben. Dále můžeme určit dobu náběhu, což je doba od počátku, kdy dojde k osvětlení součástky, až po okamžik, kdy fotorezistor dosáhne předepsané hodnoty odporu. [15]

Výhody fotorezistorů:

- 1) nízká cena
- 2) snadné použití
- 3) možnost pracovat se stejnosměrným i střídavým napětím

Nevýhody:

- 1) poměrně dlouhá doba odezvy
- 2) značná závislost na teplotě



Obr. 14 - Ukázka fotorezistoru [17]

3 IQRF

3.1 Úvod do IQRF

Jedná se o platformu určenou k bezdrátové, paketově orientované komunikaci přes RF. IQRF nabízí relativně nízké přenosové rychlosti pro přenosy menších objemů dat ovšem při velmi malé proudové spotřebě. Jeho hlavní devízou je vysoká výkonnost, bezpečnost, spolehlivost, a to vše při minimálních nákladech. Technologii IQRF můžeme použít všude tam, kde potřebujeme přenášet bezdrátově informace, dálkově monitorovat či ovládat nějaká zařízení. Výhodou bezdrátových technologií jako takových je hlavně flexibilita, kdy zaniká jakákoliv přestavba či údržba metalického vedení v případě jakékoliv změny v topologii sítě. Investice do IQRF je velice dobře návratná. Další velkou výhodou je, že při návrhu vlastního projektu s IQRF uživatel nemusí mít velké znalosti programování, datových přenosů, sítí ani dalších věcí. [18]

3.2 Technické parametry IQRF

Bezdrátový přenos se může pohybovat v pásmech 868 Mhz, 916 Mhz a 433 Mhz. Bitová rychlost pro přenos je 19,836 kb/s nebo 19,2 kb/s. Výkon, jenž je na výstupu je až do 12,5 mW nebo do 3,5 mW. Dosah přenosu je individuální na druhu prostředí, ve kterém se zrovna zařízení nachází. Pokud se nachází v budově, jedná se o desítky metrů, stovky metrů v otevřeném prostoru s malými anténami. Velkou výhodou bezdrátového přenosu je extrémně malý odběr proudu. Konkrétně 15 μ A v přijímacím módu XLP a méně než 1 μ A v režimu spánku. U IQRF můžeme rozlišit 2 komunikační módy a to:

- **non-networking:** jedná se o 2 nebo více zařízení komunikujících mezi sebou, tedy peer-to-peer. Vyslané pakety jsou dostupné pro všechny zařízení, jež se nachází v dosahu. Nejsou však přístupné žádné síťové funkce jako je routování a tak dále. Jedinou výjimkou je jednoduchá adresace.
- **networking:** jeden koordinátor může řídit až 239 nodů. Jednotlivé pakety jsou adresovány a celkově jsou podporovány všechny ostatní síťové funkce. V režimu DPA je možno pracovat pouze v modu networking. [19]

3.3 Transceiver TR-72D

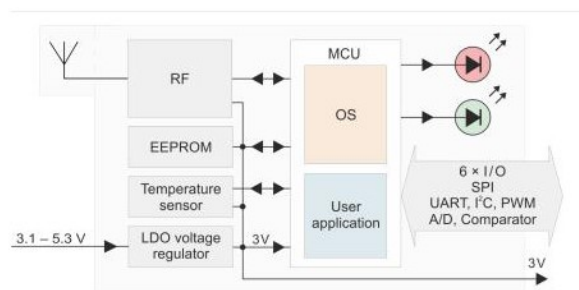
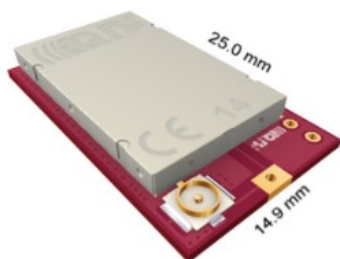
Jedná se o přijímač řady TR-72D (viz obr. 15), nástupce řady TR-52D. Má vysoký RF výkon, který je roven 8 mW. RF pásmo je 868 MHz a 916 MHz, dá se i dále softwarově konfigurovat. Dále má 6 I/O pinů. Velikost paměti EEPROM je 32 kB. Může obsahovat velice přesný snímač teploty. Nedílnou součástí jsou 2 LED diody. Napájecí napětí se pohybuje v intervalu od 3,1 do 5,3 V. Na výstupu je možno naměřit napětí o velikosti +3 V s tolerancí ± 60 mV a dodat proud o velikosti maximálně 100 mA. Přijímač má provozní teploty v intervalu od -40 do +85 °C. Přijímač může pracovat v několika režimech a to:

- režim hlubokého spánku (je podporován pouze verzí OS v4.00 nebo vyšší), kdy proudový odběr zařízení je menší než $1,7 \mu\text{A}$, ovšem všechny periferie jsou vypnuty.
- režim spánku, kdy proudový odběr zařízení je $2,3 \mu\text{A}$ a stejně jako u režimu hlubokého spánku jsou všechny periferie vypnuty.

provozní režim, kdy rozlišujeme, v jakém režimu se RF právě nachází. Pokud je RF v režimu spánku, tak jeho proudový odběr je $1,4 \text{ mA}$, pokud je v režimu ready, je jeho hodnota $2,8 \text{ mA}$.

RX režim, kdy při STD módu je proudový odběr $12,1 \text{ mA}$, LP $260 \mu\text{A}$ a XLP $18,5 \mu\text{A}$. U posledních 2 zmíněných módu záleží také na rozhraní. STD režim je také rychlejší než ostatní díky kratším časovým slotům při skoku.

TX režim, kde je proudový odběr v intervalu od $8,3$ do $21,5 \text{ mA}$ (dle výstupního výkonu RF). [19]

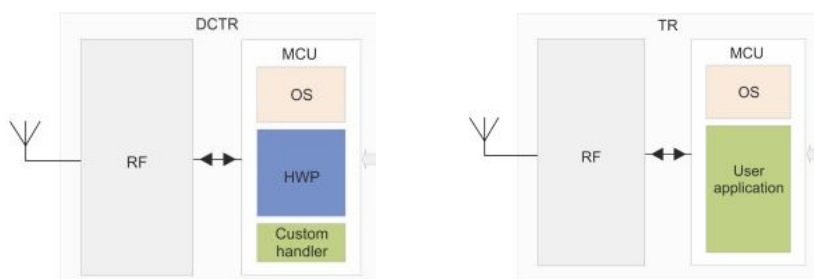


Obr. 15 – Transciever TD-72 s blokovým schématem [19]

3.5 Aplikační přístupy

3.5.1 Bez programování (DPA)

U použití DPA nemusíme použít žádné programování, ale není to podmínkou. Je možné naprogramovat tzv. Custom DPA handler. Při použití IQMESH sítě je možné použít pouze tuto možnost. [19]



Obr. 18 - Blokové schéma modelu bez programování a s programováním [19]

3.5.2 S programováním v jazyce C

Oproti verzi bez programování je tato verze daleko flexibilnější. Samotné programování probíhá přímo pod operačním systémem. Ovšem při použití se od uživatele očekávají určité vědomosti. Tuto metodu není možné použít pro IQMESH síť, ale pouze pro peer-to-peer topologii sítě. [19]

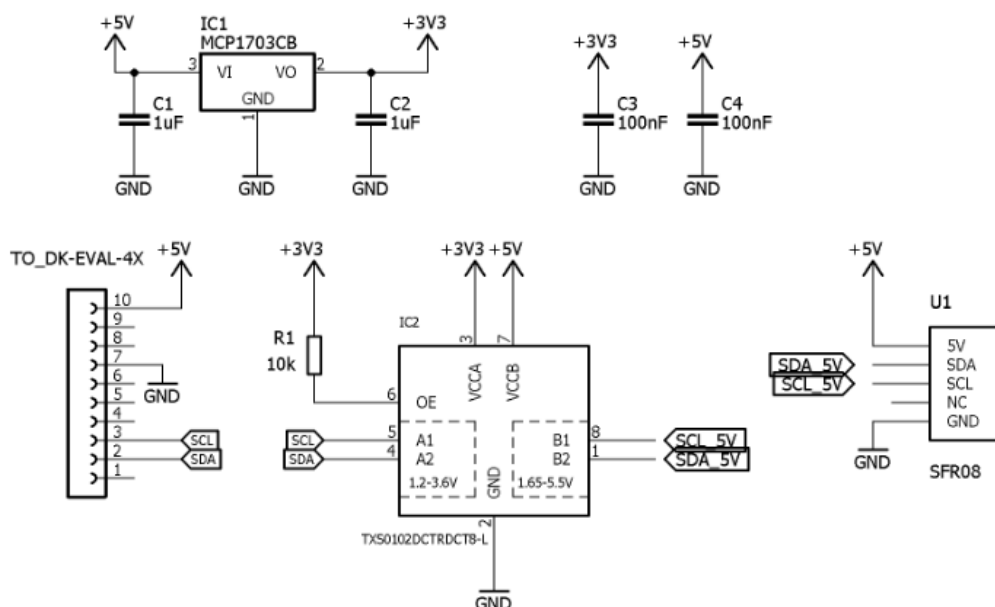
4 Návrh a realizace modulů

4.1 Návrh rozhraní mezi senzory a DK-EVAL-04A

Při návrhu šlo o to, co nejvíce ulehčit uživateli připojení jednotlivých senzorů k modulům IQRF při zachování jejich plné funkčnosti. Velikost DPS byla volena tak, aby jí bylo možno vložit do krabičky, jež má rozměry 25 x 47 mm, který má stejnou velikost, jako DK-EVAL-04A, což je modul pro rychlý vývoj obsahující slot pro vložení libovolného TR-7x modulu s 10pinovou lištou, kde každý pin odpovídá jednotlivému signálu z tohoto transceiveru. Dále byl kladen důraz na to, aby při měření jednotlivých senzorů nebyla negativně ovlivněna naměřená data.

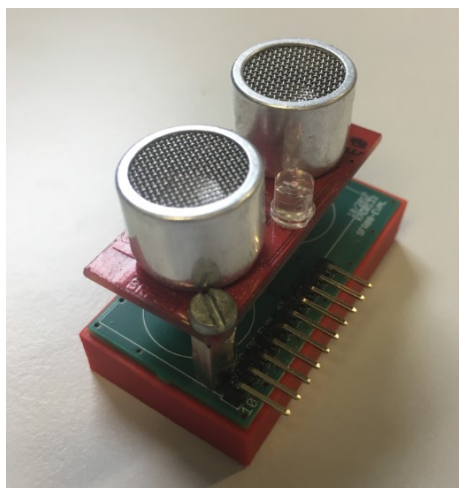
U senzoru SFR08 bylo možno díky jeho konstrukci zhotovit také jednoduché připevnění k DPS pomocí dvou šroubů, které jsou rozmístěny diagonálně v protějších rozích a u senzoru SHT21 byly vytvořeny otvory pro připevnění ochranného krytu, který by chránil senzor zejména proti nečistotám jako je prach apod. Pinová lišta odpovídá pinové liště DK-EVAL-04A, s tím že jsou zapojeny pouze ty piny, které jsou nezbytné k fungování senzorů a ulehčují uživateli připojení. Tím, že DPS je dvouvrstvá, bylo možno napájet součástky z opačné strany oproti senzorům, čímž jsou schovány v krabičce, což jim poskytuje ochranu proti vnějším vlivům okolí.

4.1.1 Modul SFR08-EVAL



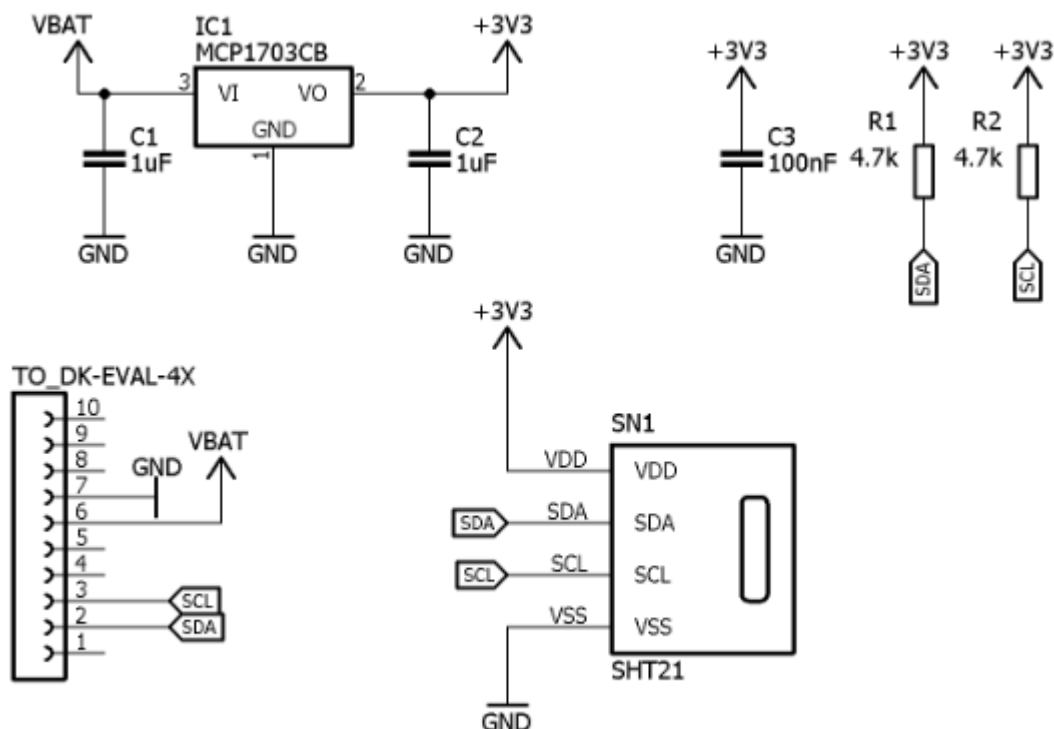
Obr. 19 - Schéma zapojení SFR08-EVAL

Tento modul (viz obr. 20) byl zhotoven pro připojení ultrazvukového senzoru SFR08 (součástka U1, viz obr. 19), který komunikuje po sériové sběrnici I2C. Senzor je napájen z externího zdroje, jelikož vyžaduje stálé napětí 5 V. Dále byl použit MCP1703CB (Low dropout voltage, součástka IC1, viz obr. 19), který vstupní napětí 5 V dokáže na výstupu převést na hodnotu 3,3 V, které slouží jako napájení pro TXS0102DCTR. Jelikož i I2C komunikace pro SFR08 zahrnuje jinou napěťovou logiku, byl použit voltage level-translator, který mění napěťovou úroveň pro komunikaci DK-EVAL-04A z 3,3 V na 5 V. K tomu byl zvolen TXS0102DCTR (součástka IC2, viz obr. 19) od společnosti Texas Instruments. Dále byly na DPS použity blokovací kondenzátory.



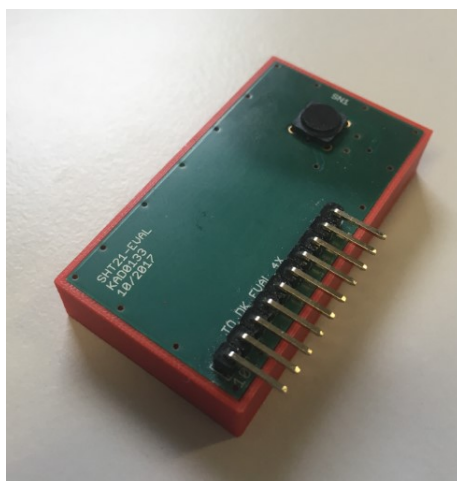
Obr. 20 - Ukázka modulu SFR08-EVAL

4.1.2 Modul SHT21-EVAL



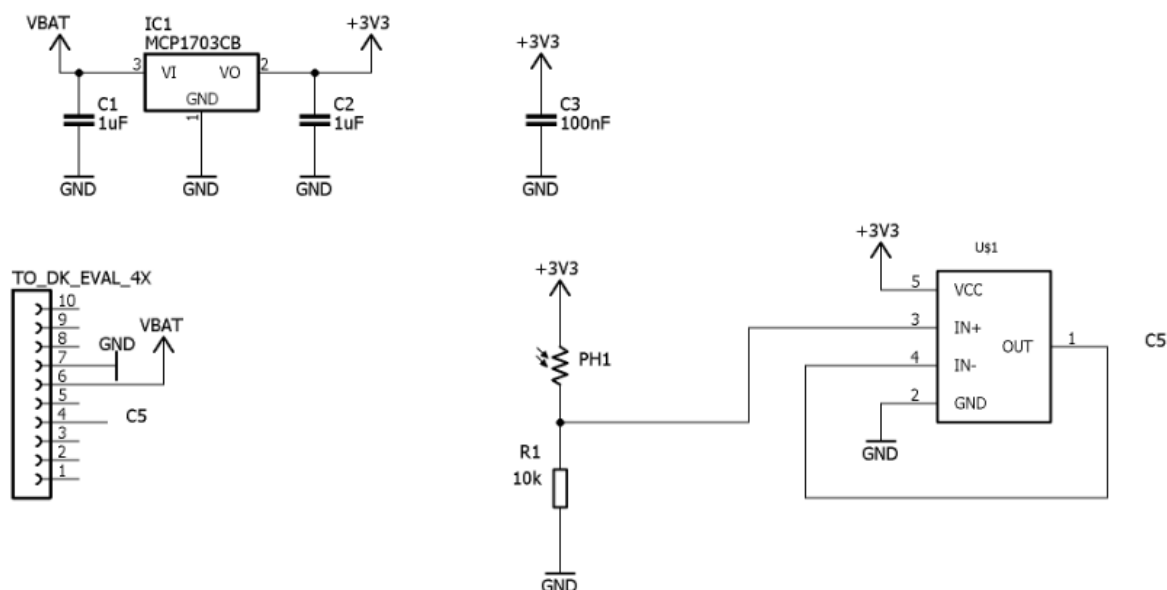
Obr. 21 - Schéma zapojení SHT21-EVAL

Modul (viz obr. 22) je určen k připojení senzoru teploty a vlhkosti SHT21 (součástka SN1, viz obr. 21) od společnosti Sensirion, který pro komunikaci s I2C modulem používá I2C sběrnici. DPS je založena na podobném principu jako u SHT21 pouze s tím, že nebylo nutné použít voltage level-translator, jelikož napěťová úroveň pro komunikaci je shodná s DK-EVAL-04A.



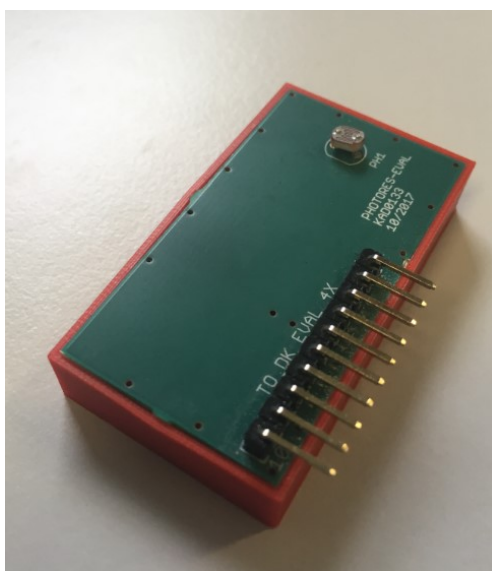
Obr. 22 - Ukázka modulu SHT21-EVAL

4.1.3 Modul PHOTORES-EVAL



Obr. 23 - Schéma zapojení PHOTORES-EVAL

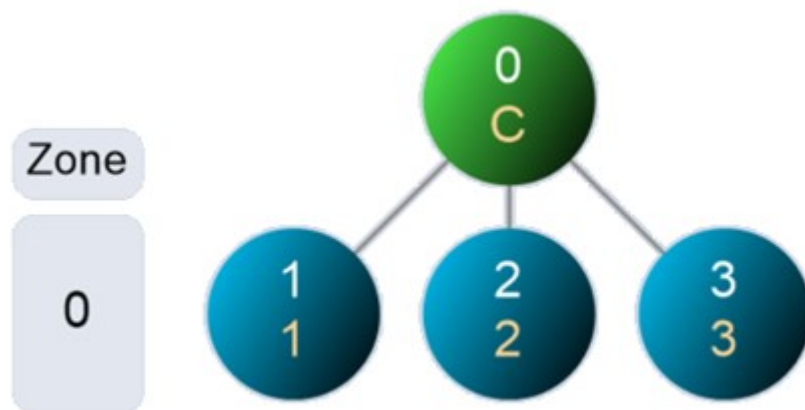
Pro návrh modulu (viz obr. 24) byl použit fotorezistor (součástka PH1, viz obr. 23), jenž je zapojen přes operační zesilovač (součástka U\$1, viz obr. 23) pracující jako sledovač, čímž výstup rezistoru impedančně oddělil. Výstup je dále zapojen na pin C5, který je připojen ke kanálu AN4 10bitového A/D převodníku obsaženého v IQRF transceiveru.



Obr. 24 - Ukázka modulu PHOTORES-EVAL

5 Vytvoření IQRF sítě

Vytvořená bezdrátová síť pracuje na frekvenci 868 MHz (střední frekvence). Proces připojení nodů ke koordinátorovi se nazývá bonding. Aby bylo možno tento krok uskutečnit, musí být v koordinátorovi síť nahrán plug-in. V tomto případě se jedná o HWP-Coordinator-STD-SPI-7Xd-V302-171116 a plug-in pro node HWP-Node-STD-7xD-V302-171116. Poté, pokud předtím nebyly nody přibondované, stačí v IQRF IDE vybrat IQMESH Network manager, kde je možnost Bound node. Po vybrání této možnosti je nutno zmáchnout na vybraném nodu tlačítko SW2, díky čemuž bude node úspěšně přibondován a bude mu udělena adresa, kterou si uživatel předem navolil, jedinou výjimku tvoří adresa 0, jelikož je rezervována pro koordinátor. Tento krok je nutno provést u každého nodu. Poté stačí kliknout na tlačítko Discovery a tím se rozkryje struktura sítě. Vytvořená síť (viz obr. 25) je vizualizována v záložce Map View.

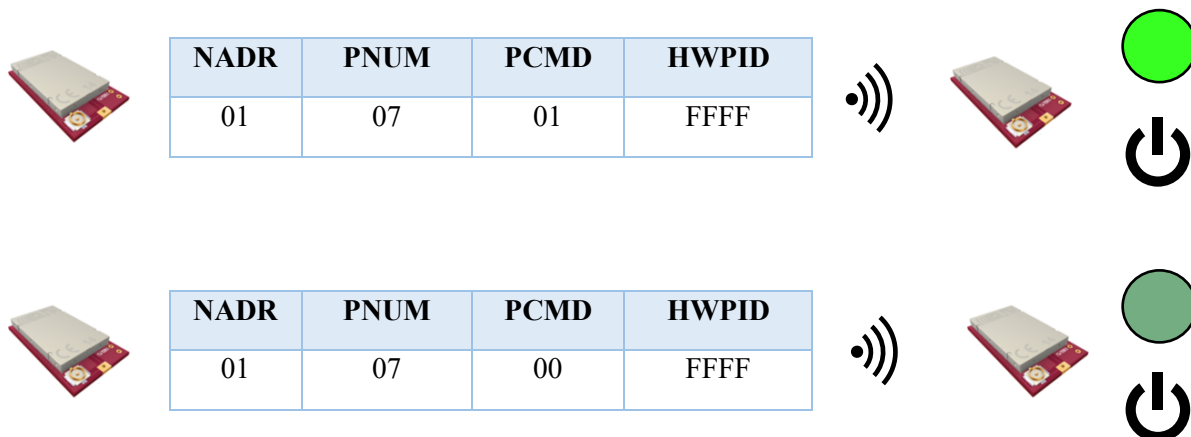


Obr. 25 – Vytvořená IQRF MESH síť

Ověřit funkčnost propojení mezi koordinátorem a nodem lze pomocí DPA testu, kdy v IQRF IDE je možno velice jednoduše složit paket, jenž se skládá z adresy nodu NADR, čísla periferie PNUM, čísla příkazu PCMD, identifikátoru hardwarového profilu HWPID a PDAT, což je datová zpráva pro daný modul a je nepovinná a uživatelsky volitelná.

Aplikace sama nabízí několik maker, kterými sama vyplní potřebná pole, například pro zapnutí a vypnutí LED diody. Uživatel volí pouze adresu nodu, jenž chce otestovat.

Ukázka zapnutí a vypnutí LED diody pro node a adresou 1:



6 Vytvoření Custom DPA handleru

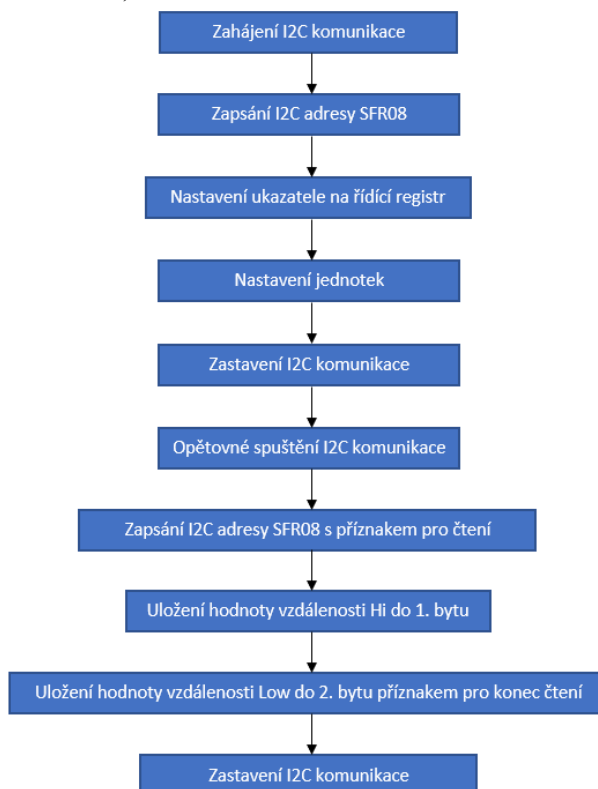
Aby jednotlivé moduly mohly komunikovat se senzory, je nutné pro každý z nich vytvořit custom dpa handler. Jelikož SHT21 a SFR08 komunikují po I2C sběrnici, byla použita šablona, která je volně dostupná na stránkách iqr.org. Pro fotorezistor bylo vycházeno z šablony pro A/D převodník. Vytvořené custom dpa handlers byly poté nahrány do jednotlivých nodů.

6.1 Custom dpa handler pro SFR08

Tento handler se skládá z jedné části, kde v souboru SFR08.c jsou obsaženy funkce, včetně hlavní smyčky programu, kde jsou vyčtena data ze senzoru a poslána na výstup.

Seznam použitých funkcí:

- `i2c_init();`
- `i2c_shutdown();`
- `i2c_waitForIdle();`
- `i2c_start();`
- `i2c_repStart();`
- `i2c_stop();`
- `i2c_read(uns8 ack);`
- `i2c_write(uns8 i2cWriteData);`



Obr. 26 - Ukázka vyčtení dat ze senzoru SFR08

Před samotnými funkcemi bylo třeba definovat unikátní HWPID pro tento senzor. Ten je dle pravidel společnosti IQRF Tech s.r.o. nutno uvést ve formátu 0xFFFF, který jsou definovaný jako uživatelský. Pro SFR08 bylo zvoleno HWPID 0x222F. I2C adresa senzoru je 0xE0.

Vyčtení dat poté probíhá v hlavním programu, viz obr. 26, kde se v hlavní smyčce nastaví příslušné hodnoty do řídicího registru po I2C sběrnici, poté se nastaví jednotka hodnotou 0x51, což znamená, že hodnota na výstupu po převedení z hexadecimálního formátu bude v centimetrech.

Poté se hodnoty uloží do 2 bytů, kde v prvním bytu je uložena hodnota Hi a v druhém bytu je uložena hodnota Low.

Po sestavení dotazu ve správném formátu se data odešlou v odpovědi, kde hodnota vzdálenosti je uložena v části PDATA.

Dotaz pro vyčtení hodnoty ze senzoru:

NADR (Low)	NADR (Hi)	PNUM	PCMD	HWPID (Low)	HWPID (Hi)
01	00	07	01	22	2F

**hexadecimální formát*

Odpověď senzoru:

NADR (Low)	NADR (Hi)	PNUM	PCMD	HWPID (Low)	HWPID (Hi)	ErrN	DPA	PDATA (Low)	PDATA (Hi)
01	00	20	80	22	2F	00	28	13	00

**hexadecimální formát*



6.2 Custom dpa handler pro SHT21

Handler je obsahuje opět jen jeden soubor s názvem SHT21.c podobně jako u předchozího případu SFR08.c.

Seznam použitých funkcí:

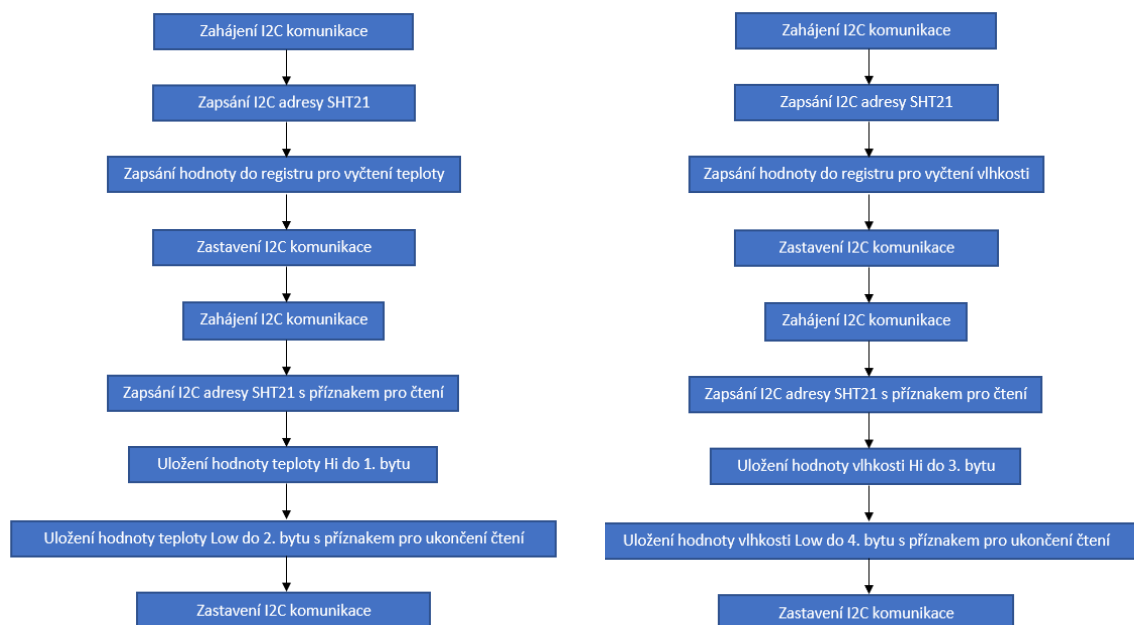
- i2c_init();
- i2c_shutdown();
- i2c_waitForIdle();
- i2c_start();
- i2c_repStart();
- i2c_stop();
- i2c_read(uns8 ack);
- i2c_write(uns8 i2cWriteData);

První rozdíl je v HWPID, kdy byla zachována unikátnost a je definován na hodnotu 0x111F. I2C adresa senzoru byla rovněž změněna na hodnotu 0x80. Poté jsou data vyčtena v hlavní smyčce. Po I2C sběrnici jsou zapsány potřebné hodnoty do uživatelského registru.

První hodnota je 0xE3, díky které je možno ze senzoru vyčíst teplotu nastavenou v stupních Celsia. Vyčítání teploty probíhá v režimu hold master, kdy senzor čeká do doby, dokud data nejsou naměřena a až poté jsou zapsána do 2 bytů.

Pro vyčtení relativní vlhkosti je nutno do uživatelského registru zapsat hodnotu 0xE5. Výsledná hodnota je poté uvedena v procentech a její vyčtení probíhá opět v režimu hold master.

Vyčtená data jsou poté pro obě měřené veličiny uloženy do 2 bytů, viz obr. 27, kde v prvním bytu je obsažena hodnota Hi, v druhém bytu hodnota Low. První 2 bity hodnoty Low jsou vymaskovány na nulovou hodnotu, jelikož se jedná o status bity a zkreslily by výslednou hodnotu při výpočtu



Obr. 27 - Princip vyčtení dat ze senzoru SHT21

Dotaz pro vyčtení hodnoty ze senzoru:

NADR (Low)	NADR (Hi)	PNUM	PCMD	HWPID (Low)	HWPID (Hi)
02	00	20	00	11	1F

**hexadecimální formát*

Odpověď senzoru:

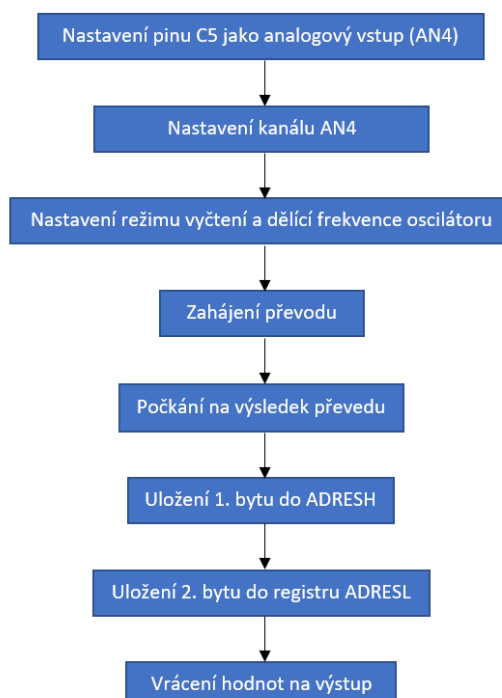
NADR (Low)	NADR (Hi)	PNUM	PCMD	HWPID (Low)	HWPID (Hi)	ErrN	DPA	PDATA (Low)	PDATA (Hi)	PDATA (Low)	PDATA (Hi)
02	00	20	80	11	1F	00	28	A8	66	C0	67

**hexadecimální formát*



6.3 Custom dpa handler pro Fotorezistor

Pro vyčtení hodnoty fotorezistoru byl použit přímo A/D převodník transceiveru TR-72D. Použit byl pouze jeden soubor PHOTORES.c. HWPID bylo nastaveno na hodnotu 0x333F.



Obr. 28 - Princip fungování programu pro vyčtení hodnoty z fotorezistoru

Pokud je vyslán paket pro vyčtení hodnoty tzn., že hodnota PCMD je rovna 0 a zároveň nenastane žádná chyba, nastaví registr ANSELA pin C5 jako analogový vstup AN4 pro A/D převodník. Registr ADCON0 nastaví kanál AN4 hodnotou 0x11. Dále se nastaví pomocí registru ADCON1 na hodnotu 0x90, která výsledek převodu vyčte zprava a nastaví dělicí frekvenci oscilátoru na hodnotu 32. GO se nastaví na hodnotu 1, čímž A/D převodník zahájí konverzi.

Poté se počká na výsledek konverze. Výsledek je uložen do 2 bytů, viz obr. 28, kde Low byte je obsažen v registru ADRESL a High byte v ADRESH.

Dotaz pro vyčtení hodnoty ze senzoru:

NADR (Low)	NADR (Hi)	PNUM	PCMD	HWPID (Low)	HWPID (Hi)
03	00	20	00	33	3F

**hexadecimální formát*

Odpověď senzoru :

NADR (Low)	NADR (Hi)	PNUM	PCMD	HWPID (Low)	HWPID (Hi)	ErrN	DPA	PDATA (Low)	PDATA (Hi)
03	00	20	80	33	3F	00	28	EB	00

**hexadecimální formát*



6.4 DPA Autoexec

Dpa autoexec je speciální program, díky němuž je možno nastavit EEPROM koordinátora pomocí poslaného paketu viz obr. 29. To znamená, že hodnota DPA v odpovědi od nodů nebude rovna síle signálu, nýbrž bude obsahovat úroveň napětí daného nodu ve voltech.

```
#pragma cdata[ __EESTART + AUTOEXEC_EEPROM_ADDR ] = \
6, PNUM_COORDINATOR, CMD_COORDINATOR_SET_DPAPARAMS, 0xFF, 0xFF, 1, \
0
```

Obr. 29 - Paket pro nastavení DPA hodnoty koordinátora sítě

Tuto hodnotu zprostředkovává funkce supplyVoltage(), která je integrální součástí operačního systému. Výsledné napětí daného nodu se poté přepočítá dle vzorce:

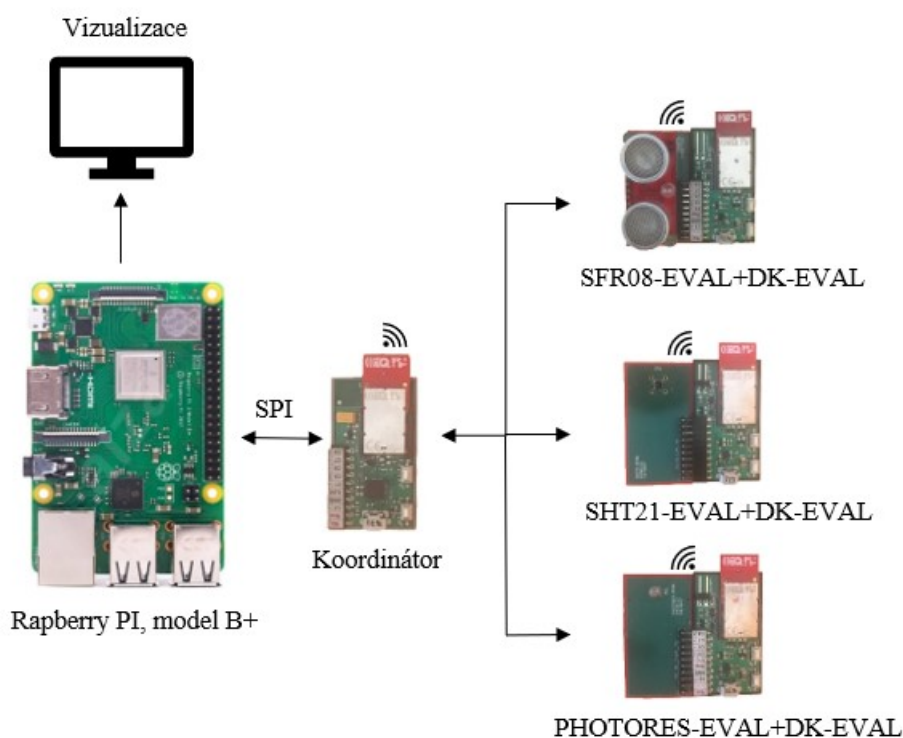
$$U(V) = \frac{261,12}{127 - DPAAvalue}$$

7 IQRF Gateway

Aby byla možná komunikace mezi Raspberry PI a IQRF moduly (viz obr. 30), bylo nutno do Raspberry PI nahrát IQRF daemon vytvořený přímo firmou Microrisc. Ten zprostředkovává komunikaci mezi nadřazenou aplikací (Node-RED) a koordinátorem pomocí MQTT protokolu, viz obr. 34, což je protokol pracující na systému M2M (machine to machine). Koordinátor je schopen komunikovat, pouze pokud v IQRF IDE je změněna USB třída z Custom device mode na CDC IQRF mode a je připojen k Raspberry PI pomocí KON-RASP-01, viz obr. 30, což je adaptér, díky kterému je schopen koordinátor komunikovat s Raspberry PI po SPI sběrnici (musí se odkomentovat v konfiguračním souboru Raspberry PI, protože je ve výchozím stavu zakázána).



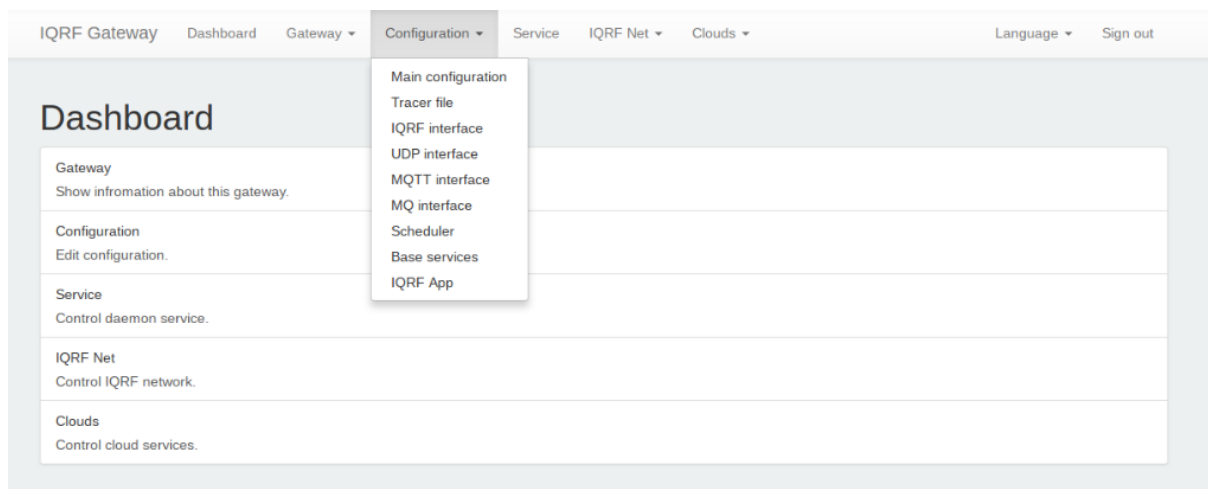
Obr. 30 - Ukázka KON-RASP-01



Obr. 31 - Struktura návrhu

Pro větší přehlednost a ulehčení další práce bylo nahráno i Webapp rozhraní (viz obr. 32), kde je možno vidět, zdali je koordinátor sítě připojen a komunikuje s nody, spravovat nastavení daemona, popřípadě posílat přímo DPA pakety nodům. Aby bylo zamezeno případnému zneužití k přístupu ke

konfiguračním souborům je nutné se do Webapp přihlásit pomocí přihlašovacího jména admin a hesla iqrif.



Obr. 32 - Ukázka Webapp včetně konfiguračních souborů

Main configuration – Zde je zobrazena konfigurační verze, cesta ke složce, kde je konfigurace uložena, WatchDog timeout, jehož hodnota se dá nastavit v ms a nastavení operačního módu, ve kterém bude gateway pracovat. Zde je nastaven operační mód, jenž se dá přepnout dále na Forwarding nebo Service, kdy se dá vzdáleně komunikovat s koordinátorem sítě přes prostředí IQRF IDE, aniž by byl fyzicky připojen k PC.

Tracer file – V prvním bodě lze vybrat, do jaké složky se budou dále ukládat logovací soubory. Také lze nastavit jakou velikost budou mít a jako poslední lze nastavit Verbosity level, kde je možno vybrat, co bude do logu zapsáno.

IQRF Interface – V tomto souboru volíme, jakým způsobem je koordinátor sítě připojen k samotnému Raspberry PI. První možnost je pomocí USB, kdy aplikace sama najde připojené zařízení v systému a automaticky ho nabídne ke zvolení. Další možnost, kterou si může uživatel zvolit je DPA Handler Timeout a komunikační mód. Ten se dělí na STD, neboli standardní režim (mnou použit) a LP, tedy Low Power. Pokud by uživatel zvolil připojení k Raspberry PI pomocí KON-RASP-01, aplikace automaticky namapuje připojené piny SPI umístěné na Raspberry PI.

UDP interface – Zde je možno povolit nebo zakázat UDP messaging, poté změnit vzdálený a lokální port.

MQTT Interface – V tomto souboru definujeme samotný protokol pro komunikaci mezi Raspberry PI a koordinátorem. Můžeme změnit jméno konfigurace, dále jí můžeme povolit či zakázat, změnit Broker address, včetně portu, který je defaultně nastaven na 1883, uživatelské ID, persistenci, QoS (quality of services).

QoS 0 – Pokud se zpráva odešle, není zde žádné potvrzení ani uložení od příjemce. Další zpráva tedy smaže předchozí.

QoS 1 – Zde odesílatel odešle zprávu a uloží si ji do doby, dokud nedostane potvrzení o přijetí od příjemce. V tomto případě je možno zprávu odeslat či přijmout i více než jednou.

QoS 2 – Tato služba zaručuje, že zpráva bude doručena pouze jednou. Je to sice nejbezpečnější metoda, nicméně je také ze všech třech nejpomalejší.

Důležité je nastavení názvu pro dotazy a odpovědi, díky čemuž můžeme komunikaci použít v námi vybraném programovacím prostředí. Komunikace jde také zabezpečit tím, že se uživatel může přihlásit pomocí přihlašovacího jména a hesla. Další parametry jsou serverového typu, tedy jako je minimální počet obnovení připojení, maximální počet obnovení připojení atd.

MQ Interface – Zde můžeme nastavit název, popřípadě službu povolit či zakázat.

Scheduler – Tady lze nastavit, jaký druh paketů je používán. Pro tuto práci jsem zvolil DPA a konkrétně RAW.

Base Services – Zde jsou obsaženy obslužné programy pro komunikaci mezi koordinátorem sítě a Raspberry PI, jako je JSON serializer apod.

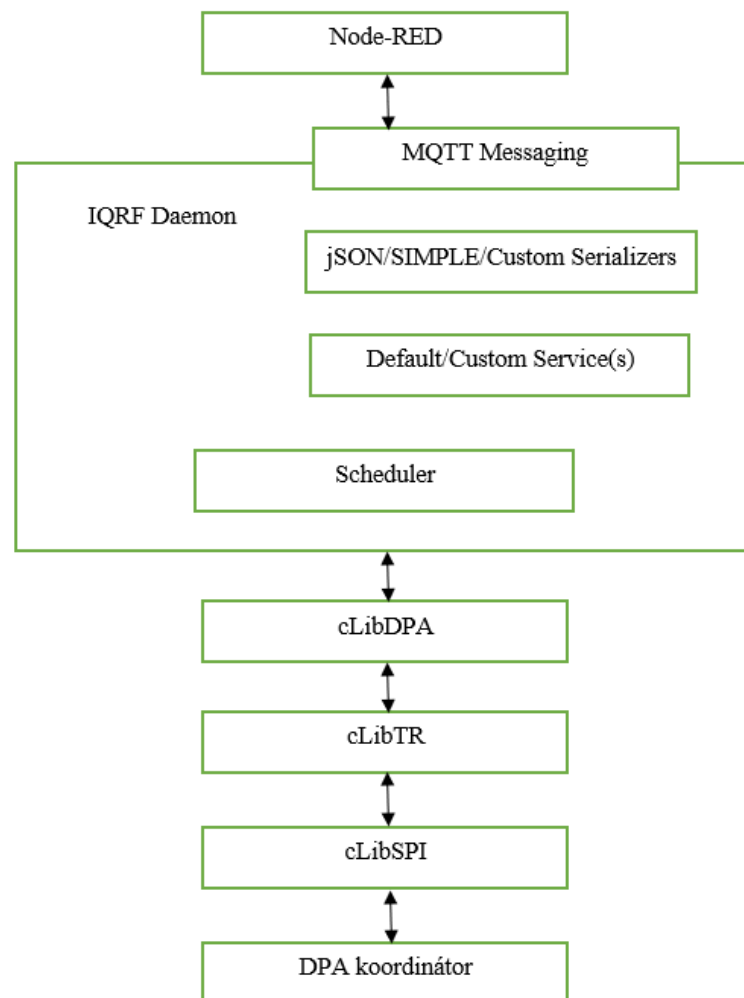
IQRF App – Konfigurace obsahuje názvy, pro vzdálené a lokální připojení včetně nastavení timeoutu, kde je nastavena defaultní hodnota na 5000.

Další možností vedle konfigurace jsou služby. Pod touto nabídkou můžeme IQRF Gateway restartovat či se podívat na status, v jakém režimu se služba IQRF-deamon aktuálně nachází.

Poslední významnou položkou je IQRF Net, kdy můžeme pomocí jednoduchého nástroje poslat paket přímo z Raspberry PI do vybraných nodů. Služba obsahuje i makra, která jsou obsažena v IQRF IDE, čímž se velice zjednodušuje práce uživateli.

Gateway Info	
Board	Raspberry Pi 3 Model B Rev 1.2
IQRF Gateway Daemon	v1.1.0 2018-03-13T21:21:15
IQRF Gateway Daemon webapp	v1.1.6 (stable - 8d2da8e4a852bb62575139f40c4a895b54dac0d5)
Hostname	raspberrypi
IP addresses	wlan0: 10.0.0.35/24, 2a00:1028:9191:d06a:a9b:aaf1:41bc:24d0/64
MAC addresses	eth0: b8:27:eb:46:59:e5 wlan0: b8:27:eb:13:0c:b0
TR info	Module type: DCTR-72D MCU type: PIC16F1938 MID: 811022BC OS: 4.02D (08B8) Supply Voltage: 3.00 V
Download diagnostics	

Obr. 33 - Zobrazení připojeného koordinátora sítě



Obr. 34 - Blokové schéma komunikace

8 Node-RED

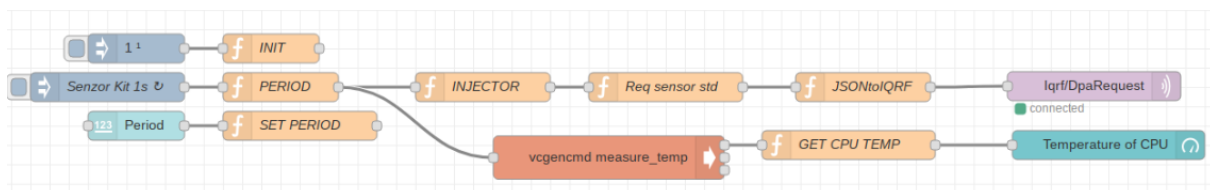
Node-RED je objektově orientovaný programovací jazyk s vizualizačními prvky, vytvořený firmou IBM. Od roku 2016 patří Node-RED pod JS Foundation a funguje na principu open source. Jeho hlavní výhodou je jednoduchost a univerzálnost, kdy našel své uplatnění zejména při programování IoT zařízení.

Základním prvkem jsou nody, kterými se myslí bloky funkcí, zprostředkování komunikací, časovače apod. Nody je možno k sobě připojit procesem, jenž se nazývá wiring. Jsou zde i bloky funkcí, do kterých uživatel může napsat svůj vlastní kód v programovacím jazyce JavaScript. Výstupem nodů jsou nejčastěji zprávy (msg.payload). Objekty je možno skládat dle libosti ve flows, jejíž činnost se vykonává paralelně a mohou spolu navzájem komunikovat.

Node-RED pracuje na zařízení ve webovém prohlížeči a dá se do něj přistupovat jak lokálně, tak i vzdáleně. To ulehčuje práci zejména v případě, pokud uživatel potřebuje programovat například na Raspberry PI, kdy grafická jednotka počítače není schopna plynulého chodu aplikace, a tudíž je více než výhodné použít právě vzdálený přístup. K vzdálenému připojení stačí, aby byl uživatel připojen v rámci jedné sítě a znal IP adresu zařízení, na kterém Node-RED pracuje (včetně portu) vzdáleného zařízení. Node-RED je obsažen v základní verzi operačního systému Raspbian Stretch verze 4.14 se základními paletami (knihovnamí).

Palety je možno jednoduše rozšířit přes uživatelskou nabídku manage pallets, kde si uživatel může stáhnout a nainstalovat požadovaný balíček nodů, jež bude potřebovat k dalšímu programování. Palety tvoří zejména sami uživatelé, jelikož se jedná o open source a díky tomu je rychlost vývoje softwaru velmi vysoká.

8.1 Poslání Requestu



Obr. 35 - Řetězec pro poslání requestu v programu Node-RED

Poslání requestu, neboli dotazu z Raspberry PI do IQRF, vychází z komunikace vytvořené partnerem IQRF JoTio Tech s.r.o. Aby paket mohl být poslán, je nutno určit, s jakou periodou chce uživatel data vyčítat.

Blok SET PERIOD (viz obr. 35), kdy pomocí celočíselného vstupu může uživatel zvolit určitou periodu (1-60 s, kde je nastaven krok po 1 s). Tato hodnota se zapíše do proměnné „p“ pomocí funkce „flow.set“, což znamená, že hodnota se změní pouze tehdy, kdy v daném flow uživatel změní hodnotu periody (je možnost použití funkce „global.set“, jež uchovává hodnotu i napříč flow, nicméně v tomto případě by toto použití bylo zbytečné). Tato proměnná dále figuruje v bloku „PERIOD“, jejíž vstupem je dále blok „inject“, který generuje každou sekundu na výstupu 1. Blok „PERIOD“ takto čítá počet 1, který vygeneruje blok „Sensor Kit 1 s“, dokud není počet tiků dělitelný nastavenou periodou beze zbytku.

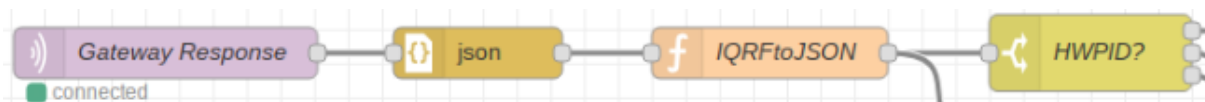
(využití funkce modulo). Pokud tato událost nastane, na výstup bloku „PERIOD“ se vygeneruje zpráva, pokud ne, blok zůstane bez výstupu.

Tento výstup je dále vstupem bloku funkce „Injector“, kdy se po obdržení vstupu vygeneruje na výstupu pole s hodnotami 1-3. Tyto hodnoty představují adresy nodů, jež jsou popsány v kapitole 7.

Složení paketu pro request probíhá v bloku funkce „Req sensor std“, kdy pomocí podmínky IF jsou jednotlivé pakety složeny pro každý node zvláště. Pokud je podmínka IF rovná 1, složí se paket pro vyčtení hodnoty z ultrazvukového senzoru, pokud 2, složí se paket pro vyčtení hodnot teploty a relativní vlhkosti z SHT21 a pokud 3, pošle se paket pro vyčtení intenzity osvětlení z fotorezistoru. Společně s paketem se zapíše i hodnota timeout (1000), která zabezpečí, že dojde ke zpracování každého paketu v pořadí od 1-3 bez jeho ztráty či poškození.

Výstup bloku funkce „Req sensor“ je vstupem bloku funkce „JSONtoIQRF“, kdy dojde ke konverzi z formátu JSON na IQRF. Takto složený paket se pošle do bloku MQTT out a za předpokladu, že je správně nastaven server včetně topicu, je paket odeslán koordinátorovi sítě přes SPI, kdy koordinátor pošle pakety do vybraných nodů. QoS je nastaven na 1. Celá tato operace se odehrává na flow s názvem „Request“

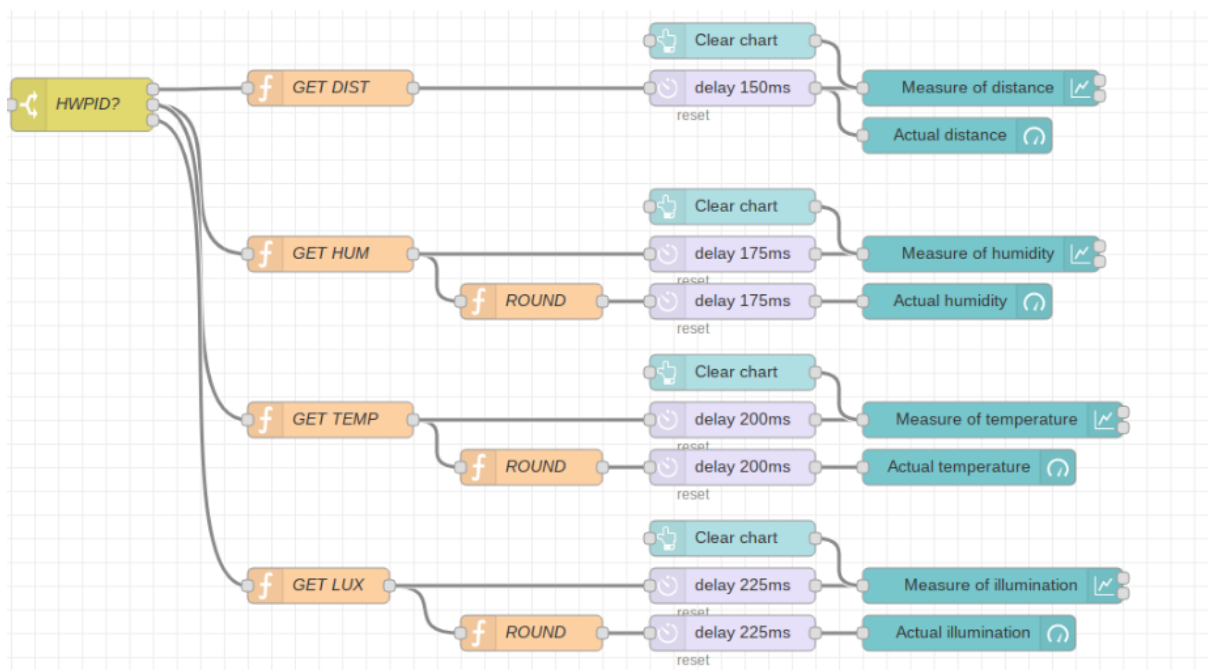
8.2 Příjmutí response



Obr. 36 - Řetězec pro přijmutí odpovědi v programu Node-RED

Pro přijmutí odpovědi z nodů přes koordinátora sítě až po Raspberry PI je nutno použít blok MQTT orientovaný jako vstup (viz obr. 36). Ten opět za předpokladu, že je správně nastaven, je schopen paket přijmout. Aby měl uživatel jistotu, že je gateway funkční, musí být pod blokem napsáno „connected“. Paket z tohoto bloku dále směřuje do bloku „json“, který slouží ke konverzi mezi JSON řetězcem a objekty. Po konverzi je přiveden na vstup bloku funkce „IQRFtoJSON“, který rozdělí paket odpovědi, tak i dotazu na jednotlivé hodnoty. Jelikož všechny hodnoty paketu mají stejný formát, není nutno řešit čísla bytu, jelikož jsou pro každý paket stejný. Dále jsou data parsována na celočíselný datový formát int, aby se s nimi mohlo dále pracovat. Jedinou výjimkou jsou PDATA, neboli hodnoty ze senzorů, kdy SHT21 vrací jako jediný senzor 4 byty. Data jak z dotazu, tak z odpovědi se uloží do msg.payload a je možno s nimi dále pracovat.

8.3 Vizualizace



Obr. 37 - Řetězec pro vizualizaci dat v programu Node-RED

Aby bylo možno vyčtená data ze senzorů vizualizovat, je třeba do Node-RED stáhnout a nainstalovat paletu „node-red-dashboard“, která obsahuje akční prvky, jako tlačítka či přepínače, včetně jednotlivých grafů nebo různých ukazatelů. Data z bloku „IQRfToJSON“ jsou tedy přivedeny na přepínač, který je pojmenován „HWPID?“ (viz obr. 37). Pomocí HWPID jsem v kapitole č. 8 jednoznačně určil, o jaký senzor se jedná. V přepínači je zapsána podmínka, pokud se hodnota HWPID rovná 0x111F, data se pošlou na výstup č. 1, když se rovná 0x222F, pošlou se na výstup č. 2 a pokud 0x333F, pošlou se na výstup č. 3.

Výstup č. 1 - výstup číslo jedna je určen pro senzor SFR08 a přiveden na vstup bloku funkce „GET DIST“. Hodnota PDAT je rozdělena pomocí funkce split na jednotlivé byty. Byty jsou poté převedeny z hexa formátu na celočíselný datový typ a uloženy do proměnné „res“. Data jsou poté před samotnou vizualizací navedena na blok „Delay“, kde proběhne zpoždění o 150 ms. Je to z důvodu, aby do vizualizace nepřistupovalo několik vstupů v jeden okamžik a vizualizace fungovala pokud možno co nejplynuleji. Z bloku „Delay“ pak data směřují do grafu a ukazatele aktuální hodnoty v podobě hladinoměru. Graf má nastavenou osu Y na minimální hodnotu na 0, maximální na 300. Tyto hodnoty jsou uvedeny v cm. Na ose X je zobrazen čas, který má nastavené automatické měřítko. Do grafu se zaznamená posledních 10 minut měření, poté dojde k přemazávání dat. Je to nastaveno z toho důvodu, jelikož pokud se do grafů začne zapisovat tisíce hodnot, grafická jednotka Raspberry PI nebude tato data schopna plynule zpracovávat a dojde k výraznému zpomalování. Hladinoměr má nastavenou minimální a maximální hodnotu shodně s grafem. Dále má uživatel možnost pomocí tlačítka „Clear chart“ vymazat veškeré zaznamenané body v grafu. Tato funkce funguje na principu, že pokud dojde ke zmáčknutí tlačítka, pošle se na vstup grafu prázdné pole hodnot.

Výstup č. 2 – tento výstup je přiřazen pro senzor SHT21. Aby bylo možné data z toho senzoru vizualizovat, je nutné z PDAT vyčíst jak hodnoty pro teplotu, tak hodnoty pro relativní vlhkost. Výstup je z tohoto důvodu přiveden na 2 bloky, jež se zpracovávají paralelně. První blok funkce se nazývá „GET HUM“, ve kterém opět pomocí funkce split dojde k rozdělení PDAT na jednotlivé byty. Jak už název napovídá, v tomto bloku půjde o vyčtení relativní vlhkosti. Tyto hodnoty jsou posílány v 2. a 3. bytu. Byty jsou opět převedeny do datového typu int a uloženy do proměnné „hum“. Nicméně aby bylo možno relativní vlhkost vyčíst je nutno použít vzorec:

$$RH(reshum) = -6 + (125 \cdot \frac{HUM}{2^{16}})$$

Výsledná hodnota po použití vzorce je uložena do proměnné „reshum“. Díky tomu je možné ji přivést na vstup grafu, přes blok „Delay“ nastavený na 175 ms, kde jsou parametry nastaveny téměř shodně, jako u vyčtení vzdálenosti, pouze s tím, že osa Y má nastavenou minimální hodnotu na 0, maximální hodnotu na 100 a jsou uvedeny v procentech.

Ukazatel zobrazující aktuální hodnotu má minimální a maximální hodnotu nastavenou opět shodně s grafem. Je nastaveno i pravidlo, že pokud je hodnota relativní vlhkosti v intervalu od 0-40, je ukazatel v žlutém poli, pokud je v intervalu 40-60, jedná se o ideální stav, a je proto v zeleném poli, pokud je více než 60, tak se ukazatel pohybuje v poli červeném. Z důvodu lepší čitelnosti a relevantnosti je hodnota zaokrouhlena na 2 desetinná místa. Pro vyčtení tepla slouží funkční blok „GET TEMP“, kde se z PDAT vybere 0. a 1. byte, který se opět převede a uloží jako int do proměnné „temp“. Opět je nutný přepočet podle vzorce:

$$T = -46,85 + (175,72 \cdot \frac{temp}{2^{16}})$$

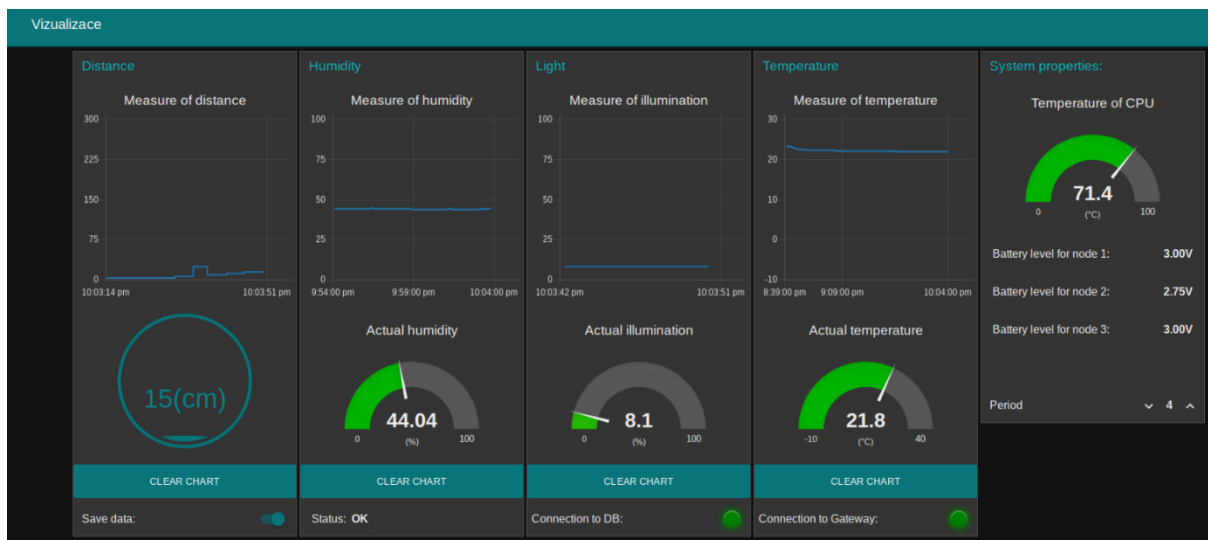
Tato hodnota je dále posílána zprávou na vstup bloku „Delay“, který má nastaveno zpoždění na 200ms. Poté zpráva dorazí na vstup grafu, který má nastavenou osu Y v intervalu -10 až +30 z důvodu lepší čitelnosti grafu. V případě nastavení plného rozsahu senzoru (-40 až +125) by demonstrace grafu při pokojové teplotě byla velmi málo čitelná. Hodnoty jsou uvedeny v stupních Celsia.

Ukazatel je poté nastaven na totožný interval jako graf, pouze s tím, že v intervalu -10 až +18 se ukazatel pohybuje ve žlutém poli, v intervalu 18-22 (což je obecně považováno za ideální teplotu v místnosti) v zeleném poli a pokud teplota přesáhne hranici 22, bude se pohybovat v poli červeném. Aby byla data opět co nejvalidnější, je výsledná hodnota zaokrouhlena na jedno desetinné místo, jelikož přesnost senzoru pro teplotu se pohybuje v rozmezí $\pm 0,3$ °C.

Výstup č. 3 – výstup z přepínače je definován pro hodnotu vyčtenou z fotorezistoru. K jejímu vyčtení z PDAT slouží funkční blok „GET LUX“, který převede hodnoty bytů do proměnné „lux“. V tomto kroku je i zakomponován vzorec, díky kterému je výsledná hodnota v procentech. Hodnotu 950 vrátil fotorezistor při použití maximálního možného zdroje světla, tudíž je brána jako 100 %.

$$RL = 100 \cdot \frac{lux}{950}$$

Před posláním výsledné zprávy do grafu musí projít blokem „Delay“, který má v tomto případě nastaveno zpoždění na 225 ms. Graf je nastaven intervalově naprosto identicky včetně ukazatele s intervalem použitý u vyčtení relativní vlhkosti. Zaokrouhlení aktuální hodnoty v ukazateli je nastaveno na jedno desetinné místo.



Obr. 38 – Celková vizualizace měřených dat v programu Node-RED

8.4 Ukládání dat

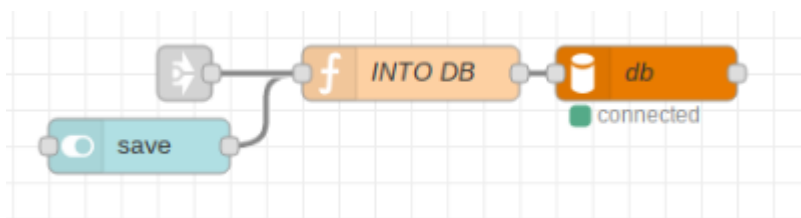
Ukládání je řešeno přes aplikaci phpMyAdmin, která je napsána v jazyce PHP a slouží pro tvorbu a správu databáze MySQL přes webový prohlížeč. Uživatel může vytvářet, editovat či filtrovat tabulky a údaje pomocí klasických SQL příkazů. Aby bylo možno phpMyAdmin spustit na Raspberry PI, je nutné společně s ním doinstalovat další potřebné aplikace. Jedna z nich je Apache, což je webový HTTP server. Při instalaci je nutné vytvořit uživatele, kterému je nutno nadefinovat uživatelské jméno a heslo a nastavit mu práva. V tomto případě pro zjednodušení byly uděleny práva „*grant all privileges“, díky čemuž vytvořený uživatel může editovat jakoukoliv před vytvořenou databázi nebo si vytvořit svou vlastní. Pokud by se při instalaci tento krok přeskočil a nedošlo by k vytvoření uživatele a přidělení práv, bylo by nutné v konfiguračním souboru phpMyAdmin povolit jako uživatele pro přihlášení root. Ten je jinak defaultně zakázán. Pokud instalace proběhne v pořádku, ve webovém prohlížeči stačí zadat „localhost“ s portem 81 (je nutné přes příkaz netstat zjistit, který port je aktuálně volný, jelikož se vlivem kolize nemusí aplikace spustit), kde po přihlášení je možné vytvořit vlastní databázi.

Pro tuto práci byla vytvořena databáze „storage_db“, která obsahuje tabulku „Quant“, kde se ukládají naměřené hodnoty z aplikace Node-RED. Společně s naměřenými veličinami obsahuje tabulka čas a datum uložení ve formátu „YYYY-MM-DD HH:MI:SS“ včetně ID (viz obr. 39), což je unikátní identifikátor každého záznamu (je zde nastavena automatická inkrementace). V Node-RED je dále potřeba přidat paletu, díky které je možno použít blok pro MySQL databázi „node-red-node-mysql“. Potom je možné do flow přidat blok z nabídky „storage“ „mysql“. Blok je funkční pouze v případě, že se správně nakonfiguruje, tzn. je potřeba správně vyplnit host adresu včetně portu, na které běží phpMyAdmin, dále uživatelské jméno a heslo uživatele, který byl vytvořen při instalaci a název databáze, do které chceme naměřené veličiny uložit. Pokud jsou tato pole vyplněna v pořádku, pod blokem se zobrazí zelený nápis „connected“.

#	Název	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Komentáře	Další	Operace
1	Time	timestamp	Ne		Ne	CURRENT_TIMESTAMP	Čas		Změnit Odstranit Primární Unikátní Klíč Prostorový Fulltext Odišné hodnoty
2	ID	int(11)	Ne	Žádná	Ne		id	AUTO_INCREMENT	Změnit Odstranit Primární Unikátní Klíč Prostorový Fulltext Odišné hodnoty
3	Distance	int(11)	Ne	Žádná	Ne		Vzdálenost		Změnit Odstranit Primární Unikátní Klíč Prostorový Fulltext Odišné hodnoty
4	Temperature	float(6,1)	Ne	Žádná	Ne		Teplota		Změnit Odstranit Primární Unikátní Klíč Prostorový Fulltext Odišné hodnoty
5	Illumination	int(11)	Ne	Žádná	Ne		Osvětlenost		Změnit Odstranit Primární Unikátní Klíč Prostorový Fulltext Odišné hodnoty
6	Humidity	float(6,3)	Ne	Žádná	Ne		Vlhkost		Změnit Odstranit Primární Unikátní Klíč Prostorový Fulltext Odišné hodnoty

Obr. 39 - Ukázka návrhu tabulky v phpMyadmin

Princip ukládání je poté založen na tom, že se čeká, až dojde k naplnění globálních proměnných pro každou hodnotu, která je měřena. Poté pokud jsou v proměnných uloženy validní hodnoty a uživatel zvolil v rozhraní nabídku uložit data, zapíše se hodnoty pomocí SQL příkazu „INSERT INTO“ do tabulky. Po odeslání tohoto příkazu se nastaví globální proměnně na hodnotu „NaN“, aby nedošlo k uložení duplicitních hodnot do tabulky pro jeden časový okamžik. Pokud by se stalo, že by senzory přestaly fungovat, do proměnných by se zapsala hodnota „NaN“ a nedošlo by ke spuštění příkazu pro uložení hodnot. Je to ošetřeno i z důvodu, že pokud by se uživatel snažil uložit hodnotu „NaN“ do tabulky, došlo by k chybě, jelikož sloupce tabulky jsou nastaveny na datový typ int nebo float a docházelo by ke kolizi datových typů. Naměřená data poté může uživatel spravovat přímo z webového prohlížeče, kde běží phpMyAdmin a může si buď data zobrazit přímo v aplikaci pomocí jednoduchého příkazu „SELECT“ nebo data exportovat například do formátu CSV, PDF, XML apod.



Obr. 40 - Řetězec pro ukládání dat do databáze v programu Node-RED

8.5 Ověření připojení

Aby uživatel měl přehled, jestli Gateway je skutečně připojená nebo jestli probíhá ukládání dat z uživatelského panelu, byly zde přidány jednoduché kontrolky, díky kterým to může jednoduše zjistit.

První status (viz obr. 41) zobrazuje, v jakém stavu se nachází Gateway. Tento parametr lze vyčíst pouze za použití bloku „status“. V tomto bloku se zaškrtně blok, jenž chceme kontrolovat. Výstup z bloku „Status“ je vstupem funkčního bloku „GATEWAY CONNECTION CHECK“. Je důležité si uvědomit, že právě blok „Status“ tvoří výjimku a nevrací na výstupu zprávu ve formátu „msg.payload“, nýbrž „msg.status“. Ve funkčním bloku je použita podmínka IF, pokud se status bloku pro Gateway rovná

„connected“, dioda je nastavena na boolovskou hodnotu „pravda“ a tím pádem kontrolka bude svítit zeleně, pokud je blok na jakémkoliv jiném stavu, kontrolka svítit nebude.



Obr. 41 - Použití STATUS1 v programu Node-RED

Druhý status blok (viz obr. 42) slouží k indikaci připojení databáze. Princip je obdobný jako při tvorbě prvního statusu, nicméně u bloku „storage“ může nastat o několik variant více. Kontrolka na výstupu bude svítit zeleně, pokud je blok ve stavu „connected“ nebo „OK“. Ve stavu „connected“ je blok v okamžik, kdy do něj nepřichází data, nicméně je správně nakonfigurován. Stav „OK“ nastává tehdy, kdy do bloku dorazí validní data a jsou dále uložena do tabulky. Pokud nastane stav „connection refused“, kontrolka zeleně svítit nebude a zůstane zhasnutá.



Obr. 42 - Použití STATUS2 v programu Node-RED

Třetí status blok slouží jako indikátor toho, jestli jsou data ukládána nebo ne. Pokud je vše v pořádku, to textového pole umístěného na panelu se vepíše „OK“, pokud by data nebyla validní a neukládala se do tabulky, zobrazí se „-“.

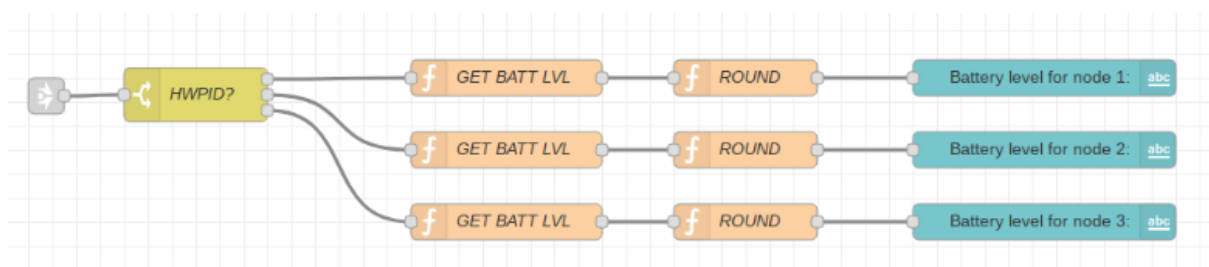


Obr. 43 - Použití STATUS3 v programu Node-RED

8.6 Napětí nodů

Jelikož komunikace mezi nody a koordinátorem sítě probíhá bezdrátově v STD režimu, je důležité, aby měl uživatel přehled o tom, v jaké úrovni se pohybuje aktuální velikost elektrického napětí baterie nodů z důvodu většího odběru než při režimu LP. Aby bylo možno vůbec elektrické napětí baterie vyčíst, bylo nutno do koordinátora sítě nahrát autoexec (viz. kapitola 8.4). V Node-RED je nejdříve nutné roztrždit jednotlivé pakety z nodů pomocí přepínače (viz obr. 44), ve kterém je nastavena podmínka přes HWPID. Takto lze vyselektovat odpovědi každého nodu a vyčíst pomocí DPA hodnoty paketu velikost elektrického napětí baterie. K tomu slouží funkční blok „GET BATT LVL“, který uloží hodnotu do proměnné „batt“. Nicméně je nutno tuto hodnotu dosadit do vzorce (viz kapitola 6.4), aby bylo možné zjistit stav baterie.

Výsledná hodnota je ve voltech, zaokrouhlena na dvě desetinná místa a je dále poslána na vstup textového pole, kde se zobrazí a uživatel, tak vidí, jestli není náhodou třeba dobítí daného nodu z elektrické sítě.



Obr. 44 - Ukázka řetězce pro vyčtení stavu baterie nodů

8.7 Teplota CPU Raspberry PI

Počítač Raspberry PI je vybaven čtyřjádrovým procesorem BCM2837, který je typu cortex-A53 o taktu 1,2 GHz. Velkou nevýhodou je, že pokud na počítači běží náročnější aplikace, tak jádro začne trpět velkým přehříváním. Hlavní důvod je, že Raspberry PI neobsahuje při koupi v základním provedení žádný aktivní ani pasivní chladič. Teplota může tedy snadno přesáhnout hranici 100 °C, aniž by začal automaticky snižovat výkon a snížil tím svoji teplotu (jako optimální se uvádí teplota při normálním běhu zařízení okolo 65 °C). Proto byl do uživatelského panelu dodán ukazatel s aktuální hodnotou teploty CPU, jelikož vizualizace naměřených dat z dlouhodobého hlediska znatelně CPU zahřívá a pokud by došlo k rapidnímu nárůstu teploty, má uživatel možnost reagovat a může buď aplikaci vypnout, popř. vypnout samotné Raspberry PI.

K vyčtení teploty CPU byl použit blok s názvem „exec“ (viz obr. 45), do kterého může uživatel zapsat libovolný systémový příkaz a jeho výstup dále použít pro další zpracování. V tomto případě pro teplotu CPU byl použit příkaz „vcgencmd measure_temp“. Funkce vrací teplotu s doprovodným textem, tudíž bylo nutno použít ve funkčním bloku s názvem „GET CPU TEMP“ funkci substring, jež je schopna z výstupního textu vybrat právě jen hodnotu teploty. Teplota je uvedena ve stupních Celsia a přivedena na ukazatel, který má nastavený interval 0-75 na zelenou barvu, interval 75-85 na žlutou a 85-100 na červenou.



Obr. 45 - Vyčtení teploty CPU Raspberry PI v programu Node-RED

9 Závěr

Účelem této bakalářské práce bylo realizovat vzdálené měření neelektrických veličin pomocí IQRF technologie s možností naměřená data vizualizovat a archivovat na počítači Raspberry PI. Jako měřené veličiny byly zvoleny vzdálenost, teplota, vlhkost a intenzita osvětlení.

Za tímto účelem byly zhotoveny jednoduché moduly pro připojení senzorů k nodům, pomocí kterých jsou naměřená data odesílána do koordinátora sítě.

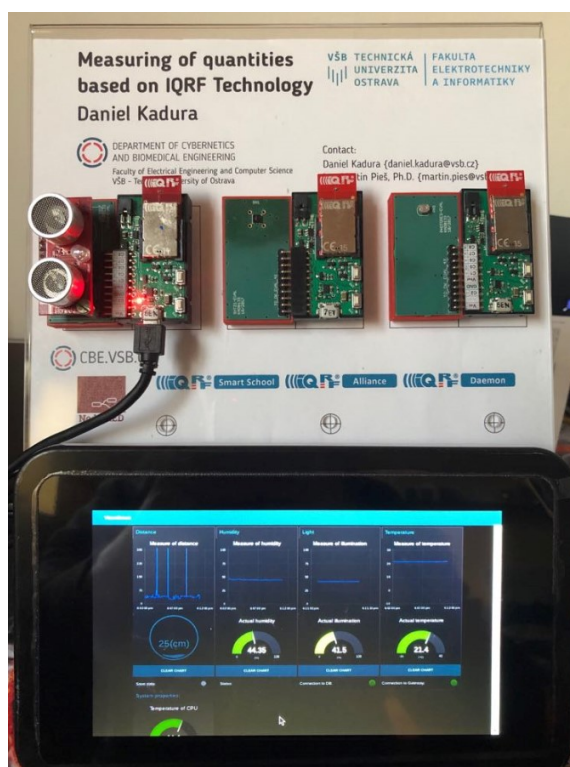
Topologie sítě je typu MESH, vytvořená v prostředí IQRF IDE. Pakety pro dotaz a odpověď odpovídají formátu DPA protokolu pro komunikaci.

Aby bylo možno data ze senzorů vyčíst, byly vytvořeny custom DPA handlers pro každý node, jež vytváří komunikaci mezi nodem a senzorem přes I2C sběrnici nebo A/D převodník.

Data jsou poté vizualizována a ukládána na jednodeskový počítač Raspberry PI pomocí aplikace Node-RED, kde bylo vytvořeno jednoduché prostředí, kde má uživatel přehled o měřených hodnotách z hlediska delšího časového období díky grafům, včetně aktuálních hodnot zobrazených na ukazatelích, dále má přehled o tom, jestli komunikace pomocí gateway je v pořádku a jestli jsou data ukládána nebo ne.

Data jsou ukládána do MySQL databáze, jenž zprostředkovává aplikace phpMyAdmin, kde má vytvořený uživatel nastavena nejvyšší oprávnění (povoleny atributy pro čtení i zápis). Tyto hodnoty je pak možno exportovat do dalších formátů, aby mohly být dále zpracovány.

Tento demonstrační měřicí aparát byl vložen na panel (viz obr. 46) a bude sloužit dále k výukovým účelům.



Obr. 46 - Vytvořený panel

Pokud bych měl shrnout nedostatky, tak vizualizační a archivační proces je pro Raspberry PI velmi výkonnostně zatěžující a je proto lepší volit výkonnější jednodeskový počítač např. UP board od společnosti AAEON. Další kritérium, které by se dalo zlepšit, je použití vizualizačního prostředí GRAFANA oproti Node-RED, která nabízí daleko přehlednější grafy.

Zdroje

- [1]. Teplo a měření [online]. [cit. 2017-08-12].
Dostupné z: http://artemis.osu.cz:8080/artemis/uploaded/199_3_6%20Teplo-mereni.pdf
- [2]. Kontaktní měření teploty [online]. [cit. 2017-08-12].
Dostupné z: http://www.umel.feec.vutbr.cz/bmms/projekty_2004/Benes/index.html
- [3]. Termoelektrické senzory [online]. [cit. 2017-08-12].
Dostupné z: http://www.kom.tul.cz/to/sdilene/teplotni_senzory.pdf
- [4]. Princip bezdotykového měření teploty [online]. [cit. 2017-08-12].
Dostupné z: <https://automatizace.hw.cz/rehled-principu-el-mereni-teploty-2-dil-bezdotykov>
- [5]. Kvantové senzory IR záření [online]. [cit. 2017-08-12].
Dostupné z: <https://eluc.kr-olomoucky.cz/verejne/lekce/1581>
- [6]. Odporové snímače polohy [online]. [cit. 2017-08-12].
Dostupné z: <http://automatizace-issnp.wz.cz/Soubory/odpor%20poloha.pdf>
- [7]. Kapacitní snímač polohy [online]. [cit. 2017-08-12].
Dostupné z: http://www.umel.feec.vutbr.cz/~adamek/uceb/DATA/s_4_1.htm
- [8]. Indukční snímače [online]. [cit. 2017-08-12].
Dostupné z: <https://automatizace.hw.cz/komponenty-mereni-a-regulace/indukcni-snimace-priblizeni-obecny-popis.html>
- [9]. Optické snímače [online]. [cit. 2017-08-12].
Dostupné z: <http://www.edumat.cz/texty/poloha.pdf>
- [10]. Optické snímače polohy [online]. [cit. 2017-08-12].
Dostupné z: <http://student-spst.kvalitne.cz/eza/poloha.pdf>
- [11]. Ultrazvukový snímač polohy [online]. [cit. 2017-08-12].
Dostupné z: <http://automatizace-issnp.wz.cz/Soubory/Snimace%20polohy%20ultrazvukove.pdf>
- [12]. Měření polohy [online]. [cit. 2017-08-12].
Dostupné z: <http://student-spst.kvalitne.cz/eza/poloha.pdf>
- [13]. SFR08, ultrazvukový snímač [online]. [cit. 2017-08-12].
Dostupné z: <https://www.robotgear.com.au/Product.aspx/Details/287-Devantech-SRF08-Ultrasonic-Sonar-Range-Finder>
- [14]. Princip luxmetru [online]. [cit. 2017-08-12].
Dostupné z: <http://www.powerwiki.cz>
- [15]. Fotorezistory [online]. [cit. 2017-08-12].
Dostupné z: <https://eluc.kr-olomoucky.cz/verejne/lekce/610>
- [16]. Fotorezistory a jejich fungování [online]. [cit. 2017-08-12].
Dostupné z: http://elektross.gjn.cz/soucastky/zadny_prechod/fotorezistor.html
- [17]. Obrázek fotorezistoru [online]. [cit. 2017-08-12].
Dostupné z: <http://www.joyta.ru/7603-fotorezistor-osnovnaya-informaciya/>

- [18]. O platformě IQRF [online]. [cit. 2017-08-12].
Dostupné z: <https://vyvoj.hw.cz/rf/male-chytre-ceske-bezdratova-platforma-iqrf.html>
- [19]. IQRF [online]. [cit. 2017-08-12].
Dostupné z: <https://iqrf.org>
- [20]. Quick_Start_Guide_IQRF_170821 [online]. [cit. 2019-02-02].
Dostupné z: <https://iqrf.org>

Seznam příloh:

Příloha A:	Postup spuštění aplikace	I
------------	--------------------------------	---

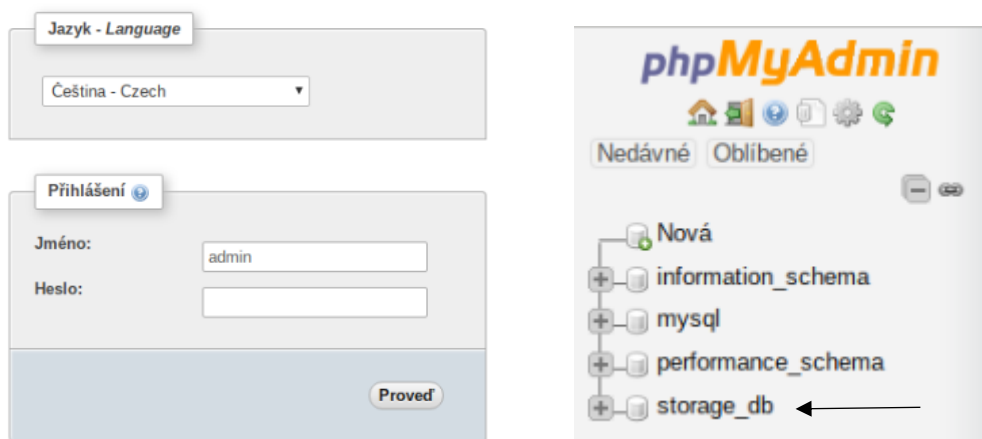
Součástí BP je CD/DVD.

Adresářová struktura přiloženého CD/DVD:

- A_DPS – Schémata modulů vytvořených v programu Eagle
- B_IQRF – Jednotlivé Custom DPA Handlers pro každý modul + autoexec, včetně softwarových pluginů a konfiguračního souboru
- C_Node-RED – Vytvořená vizualizace v programu Node-RED
- D_Stojan – Obrázek použitý na pozadí stojanu
- E_Obrazová_dokumentace – Obrázky a fotografie řešení

Postup spuštění aplikace

- 1) Zasunout jumperů na vývojové moduly DK-EVAL-04A.
- 2) Připojit koordinátora pomocí KON-RASP-01 k Raspberry PI + připojit napájení na node, kde je zapojen SFR08.
- 3) Spustit aplikaci IQRf daemon zadáním adresy localhost do webového prohlížeče (login: admin, heslo:iqrf) a zkontrolovat, jestli se načetla data o koordinátoru (viz obr. 33).
- 4) Spustit Node-RED ve webovém prohlížeči (Chromium) (Node-RED se zapne sám po zapnutí Raspberry PI a tudíž není nutné psát příkaz k jeho spuštění), který je na adrese 127.0.0.1:1880 a provést deploy pro nahrané flows.
- 5) Pro zapnutí vizualizace je nutné napsat do adresy 127.0.0.1/1880/ui
 - Pro správu databáze je nutné přihlášení do phpMyAdmin (login admin, heslo iqrf), který je na adrese localhost:81/phpMyAdmin.
 - Zde vybrat databázi storage_db, kde jsou uložena data (viz obr. 47).



Obr. 47 - Přihlášení a zvolení databáze v aplikaci phpMyAdmin